CITY
UNIVERSITY OF LONDON
EST 1894

A Novel Human Visual Psychophysics Based Approach to Distinguish Between Human Users and Computer Robots

A Thesis Submitted to

City, University of London, Department of Electrical and Electronic Engineering

In Full Fulfilment of the Requirement for the Degree

Doctor of Philosophy (PhD) in

Electrical and Electronic Engineering

By

Seyedmohammadreza Saadatbeheshti

December 2017

Declaration

I hereby declare that except where specific reference is made to the work of others, the contents of this dissertation are original and have not been submitted in whole or in part for consideration for any degree or qualification in this, or any other university. This dissertation is my own work and contains nothing which is the outcome of work done in collaboration with others, except as specified in the text. This dissertation contains fewer than 56,000 words including bibliography, footnotes, tables, and equations. It also contains over 100 figures and 20 tables.

Acknowledgment

Firstly and most importantly, I would like to warmly thank my supervisor, Professor Muttukrishnan Rajarajan, for his constant and inspiring support and exceptional expertise, which have guided this project to completion. This work would not have been possible without his diligent supervision, and I am deeply grateful to him. I would also like to take this opportunity to thank my former supervisor, Professor Panos Liatsis, for his continuous support during my PhD project, and for also initially inspiring my research.

I am also equally thankful to my family, especially my lovely wife Golriz Beheshti for her constant encouragement, support, and patience. I would like to thank her for making our home a pleasant and calm environment, which has given me ample opportunity to focus on my studies over the years.

Finally, I would like to thank my parents, particularly my darling mother for her great support and inspiration from thousands of miles away during my studies in London. She taught me the importance of love and perseverance, which has made me very resilient in my PhD journey. I am so thankful to her and I certainly could not have completed this project without her support.

List of Notations

TT	Turing Test
CAPTCHA	Completely Automated Public Turing test to tell Computers and Humans Apart
OCR	Optical Character Recognition
HCR	Human Character Recognition
GWAP	Games With A Purpose
iCAPTCHA	Interactive CAPTCHA
HIP	Human Interaction Proofs
IMCA	Instant Messenger CAPTCHA Attack
CRT	Cathode Ray Tube
LCD	Liquid Crystal Display
SM	Sensory Memory
STM	Short-Term Memory
LTM	Long-Term Memory
FPS	Frame Per Second
PPF	Pixel Per Frame
ROD	Random Output Data
SFS	Single Frame Scenario
CFS	Consecutive Frames Scenario
ID	Identification

IG	Imitation Game
POV	Persistence of Vision
IM	Iconic Memory
ASP	Automatic Puzzle Solver
GPS	Global Positioning System
LSB	Least Significant Bit
OSR	*Object Sampling Rate*
BNR	*Background Noise Rate*
MSB	Most Significant Bit
VICAP	Visual Integration CAPTCHA
ROD	Random Output Data
RIL	Recognition Improvement Level
SSM	Single Stage Model
MSM	Multi-Stage Model
ORO	Original-to-Random Output
HRSR	Human Recognition Success Rate
CRSR	Computer Recognition Success Rate
E2E	End2End Authentication
SSS	Single Stage Scenario
MSS	Multi-Stage Scenario
CGA	CAPTCHA-Generator Application
CTA	CAPTCHA-Test Application
CUEP	CAPTCHA User Experience and Performance

Abstract

Demand for the use of online services such as free emails, social networks, and online polling is increasing at an exponential rate. Due to this, online service providers and retailers feel pressured to satisfy the multitude of end-user expectations. Meanwhile, automated computer robots (known as ‘bots’) are targeting online retailers and service providers by acting as human users and providing false information to abuse their service provisioning. CAPTCHA is a set of challenge/response protocols, which was introduced to protect online retailers and service providers from misuse and automated computer attacks. Text-based CAPTCHAs are the most popular form and are used by most online service providers to differentiate between human users and bots. However, the vast majority of text-based CAPTCHAs have been broken using Optical Character Recognition (OCR) techniques and thus, reinforces the need for developing a secure and robust CAPTCHA model. Security and usability are the two fundamental issues that pose a trade-off in the design of a CAPTCHA. If a CAPTCHA model were too difficult for human users to solve, it would affect its usability, but making it easy would risk its security.

In this work, a novel CAPTCHA model called VICAP (Visual Integration CAPTCHA) is proposed which uses trans-saccadic memory to superimpose a set of fleeting images into a uniform image. Thus, this will be creating a meaningful picture of the object using the sophisticated human visual system. Since the proposed model is based on this unique ability of humans, it is logical to conclude that none of the current computer recognition programmes has the ability to recognise and decipher such a method. The proposed CAPTCHA model has been tested and evaluated in terms of usability and performance in laboratory conditions, and the preliminary results are encouraging. As a result of this PhD research, the proposed CAPTCHA model was tested in two scenarios. The first scenario considers the traditional setup of a computer attack, where a single

frame of the CAPTCHA is captured and passed on to the OCR software for recognition. The second case, implemented through our CAPTCHA-Test Application (CTA), uses prior knowledge of the CAPTCHA design. Specifically, a number of frames are individually captured and superimposed (or integrated) to generate output images as a single image using the CTA and then fed into the OCR programme. The second scenario is biased because it also requires prior knowledge of the time interval (ISI) to be used in the integration process. When the time interval is set to a value higher than the optimal ISI, there is insufficient information to complete the CAPTCHA string. When the time interval for integration is set to a value lower than the optimal one, the CAPTCHA image is saturated due to the uniform nature of the noise process used for the background.

In order to measure the level of usability of our proposed VICAP model, a user evaluation website was designed to allow users to participate in the proposed VICAP model. This evaluation website also enabled participants to compare our proposed VICAP model with one of the current popular Google CAPTCHA models called ReCAPTCHA. Thus, to ensure the usability of the proposed CAPTCHA model, we set the threshold for the ORO (Original to Random Output Data) parameter at 40%. This ensured that our CAPTCHA strings would be recognised by human observers at a rate of 100%. In turn, when examining the robustness of our VICAP model to computer programme attacks, we can observe that for the traditional case of OCR recognition, based on a single-frame scenario, the Computer Recognition Success Rate (CRSR) was about 0%, while in the case of a multi-frame scenario, the CRSR can increase to up to 50%. In the unlikely scenario of an advanced OCR software attack, comprising of frame integration over an optimal time interval (as described above), the robustness of the VICAP model for the multi-frame sequence reduces to 50%. However, we must stress that this latter scenario is unfairly biased because it is not supported by the capabilities of present state-of-the-art OCR software.

Publications

The following publications have been produced and presented as results of this PhD research project:

Journal Publication:

1. Seyed Mohammad Reza Saadat Beheshti, Panos Liatsis, Muttukrishnan Rajarajan, A CAPTCHA model based on visual psychophysics: Using the brain to distinguish between human users and automated computer bots, *Computers & Security*, Volume 70, 2017, Pages 596-617, ISSN 0167-4048

International Conferences:

1. Seyed Mohammad Reza Saadat Beheshti & P. Liatsis, "CAPTCHA Usability and Performance, How to Measure the Usability Level of Human Interactive Applications Quantitatively and Qualitatively?" *2015 International Conference on Developments of E-Systems Engineering (DeSE), Dubai, 2015*, pp. 131-136

2. Seyed Mohammad Reza Saadat Beheshti & P. Liatsis, "VICAP: Using the mechanisms of trans-saccadic memory to distinguish between humans and machines," *2015 International Conference on Systems, Signals and Image Processing (IWSSIP), London, 2015*, pp. 295-298

3. Seyed Mohammad Reza Saadat Beheshti & P. Liatsis, "How humans can help computers to solve an artificial problem?" *2015 International Conference on Systems, Signals and Image Processing (IWSSIP), London, 2015*, pp. 291-294

Table of Contents

ACKNOWLEDGMENT 3

LIST OF NOTATIONS 4

ABSTRACT 6

PUBLICATIONS 8

LIST OF FIGURES 12

LIST OF TABLES 21

LISTINGS 23

1 INTRODUCTION 24

1.1 HUMAN INTERACTIVE PROOFS (HIPS) AND CAPTCHAS 24

1.2 RESEARCH OBJECTIVES 28

1.3 CONTRIBUTIONS 30

1.4 OUTLINE OF THESIS 30

2 STATE-OF-THE-ART IN HUMAN INTERACTIVE PROOFS AND CAPTCHAS 33

2.1 INTRODUCTION 33

2.2 THE TURING TEST AND IMITATION GAME 33

2.2.1 Standard Interpretation 34

2.3 CAPTCHA CONCEPT 36

2.4 HOW DOES A CAPTCHA WORK? 37

2.5 CAPTCHA APPLICATIONS 38

2.6 THE BASIC CATEGORISATION OF THE CAPTCHAS 40

2.7 SECURITY OF CAPTCHAS 42

2.7.1 Automated Character Recognition Software Attacks (OCR) 42

2.7.2 Manual Third-Party Human Attacks 51

2.7.3 Comparison Analysis of Automated Attacks vs. Manual Attacks 52

2.8 COMPARISON ANALYSIS OF OCR AND NON-OCR-BASED CAPTCHAS 53

2.9 COMPARISON ANALYSIS OF THE CURRENT CAPTCHA MODELS 54

2.10 CAPTCHA USABILITY AND PERFORMANCE 56

2.11 CONCLUSIONS 62

3 A VISUAL PSYCHOPHYSICS APPROACH TO THE PROPOSED CAPTCHA MODEL 63

3.1 INTRODUCTION 63

3.2 PERSISTENCE OF VISION AND CAPTCHA 63

3.3 PERSISTENCE OF VISION AND THE FILM INDUSTRY 65

3.4 THE EFFECT OF PERSISTENCE OF VISION IN TEMPORAL INTEGRATION 66

3.5 TRANS-SACCADIC VISUAL INTEGRATION TECHNIQUE 68

3.5.1 Single Stage Scenario (SSS) 69

3.5.2 Multi Stage Scenario (MSS) 71

3.6 CONCLUSIONS 72

4 PROPOSED VISUAL INTEGRATION CAPTCHA MODEL (VICAP V.1) 73

4.1 INTRODUCTION 73

4.2 INTRODUCING VISUAL INTEGRATION CAPTCHA (VICAP) VERSION.1 73

4.3 CAPTCHA-GENERATOR APPLICATION OVERVIEW 74

4.3.1 Binarisation and Bitmap Conversion 80

4.3.2 Object Sampling Rate 82

4.3.3 Adding Background Noise to the VICAP Image 84

4.4 DEVELOPMENT OF THE CAPTCHA-TEST APPLICATION 95

4.4.1 CAPTCHA-Test Application Overview 95

4.4.2 8-bit/pixel, Greyscale Format Output Image 98

4.4.3 1-bit/pixel, Black and White Output Image 102

4.5 VICAP V.1 EXPERIMENTAL SIMULATION OUTPUT RESULTS 102

4.5.1 5 Frames Superimposed Simulation Results 103

4.5.2 10 Frames Superimposed Simulation Results 104

4.5.3 The Impact of the Variable OSR vs. Changes in the Background Noise Level on CRSR ... 106

4.6 INTEGRATING VICAP INTO WEBPAGES 109

4.7 PROBLEM ASSOCIATED WITH THE VICAP V.1 112

4.8 CONCLUSIONS 115

5 ORIGINAL-TO-RANDOM FRAMES CAPTCHA MODEL (VICAP V.2) 116

5.1 INTRODUCING ORIGINAL-TO-RANDOM OUTPUT DATA (ORO) PARAMETER 116

5.2 VICAP V.2 ATTACKS AND SECURITY ANALYSIS 119

5.2.1 Optical Character Recognition Techniques and Vertical Segmentation 119

5.2.2 Third Party Human Attacks 121

5.2.3 Dictionary-Based Attacks 122

5.2.4 Text-Recognition Algorithm Working on Visual Traces 122

5.3 VICAP V.2 SECURITY AND ROBUSTNESS EXPERIMENTAL RESULTS AND DISCUSSION 124

5.3.1 Experiment 1: Computer Recognition Success Rate (CRSR) 124

5.3.2 Experiment 2: Human Recognition Success Rate (HRSR) 138

5.3.3 ORO Analytical Comparison 142

5.4 VICAP V.2 USABILITY AND PERFORMANCE EXPERIMENTAL RESULTS 145

5.4.1 Comparison of VICAP and Current CAPTCHA Model in terms of Usability and Performance 148

5.4.2 Diversions in Human Perception 159

5.5 CONCLUSIONS 161

6 CONCLUSION AND FUTURE WORK 162

6.1 CONCLUSION 162

6.2 FUTURE DIRECTION OF CAPTCHA RESEARCH 169

BIBLIOGRAPHY 170

List of Figures

Figure 1.1: Example of Text-based HIPs. It should be noted that letters in this CAPTCHA are deformed and the lines that have been added create problems for an OCR system. [3] 25

Figure 1.2: An example of ReCAPTCHA by google. The image shows two distorted words and the user is required to decipher and recognise [6] 26

Figure 1.3: Changing of the function of HIPs according to its difficulty level for computer programmes and human users [3] 27

Figure 2.1: First version of the IG. The role of interrogator is to identify which one of the two players is a woman [9]. 34

Figure 2.2: Second version of IG known as the TT. As it is shown in this example, the role integrator is to identify which of the two players is machine and which one is human. [9] 34

Figure 2.3: TT falls into the intersection area between typical human behaviour and a computer's intelligent behaviour [10]. 35

Figure 2.4: An example of Facebook CAPTCHA. The user is required to recognise all the letters displayed in the distorted image in order to access the website [11]. 36

Figure 2.5: General framework for the CAPTCHA authentication process. 38

Figure 2.6: Example of a Text-based CAPTCHA used by Yahoo! It can be noticed that the text is reformed in order to make it difficult for the OCR programmes to recognise [14]. 39

Figure 2.7: The basic categorisation of CAPTCHAs. As the figure shows there are three main categories of CAPTCHA according to their nature with examples of each. 41

Figure 2.8: The graph shows various stages involved in the OCR recognition process. 43

Figure 2.9: Converting the CAPTCHA image into greyscale mode [27]. 44

Figure 2.10: CAPTCHA image after line removal phase. As the image shows all the vertical and horizontal lines have been removed from the background [27]. 45

Figure 2.11: CAPTCHA image after the dot removal phase [27]. 47

Figure 2.12: All the letters are being scaled after segmentation phase [27]. 47

Figure 2.13: The process of converting a segmented character into a binary image, then converting it to I-matrix [27]..47

Figure 2.14: The Gimpy CAPTCHA image shown in the image has got mesh background with diagonal uniform shape instead of straight lines...48

Figure 2.15: the CAPTCHA image is rotated by β degree in order to make the mesh background straight vertical and horizontal lines. 48

Figure 2.16: The final rendered output CAPTCHA image after applying “dot-removal” algorithm. As it is shown in the image, the background is removed using vertical and horizontal line removal algorithm...49

Figure 2.17: The CAPTCHA image is rotated by -β degree to bring it to the first position. As the image shows, the final characters can be easily recovered after applying segmentation and pattern matching algorithm. 49

Figure 2.18: An Example of a CAPTCHA model in use at Yahoo! as it can be seen from the image, the text is printed in black against colourful background [28]. 50

Figure 2.19: The final output rendered image. The background is completely removed using “colour matching” techniques and the black text is only remaining in the image.50

Figure 2.20: The figure shows after the background noise is removed, the text has been recognised using segmentation techniques. 50

Figure 2.21: An example of an unreadable CAPTCHA due to the extreme level of distortion [39]... 58

Figure 2.22: Examples of over ambiguous CAPTCHA characters. 58

Figure 2.23: MSN CAPTCHA where it is not clear whether the first letter is a ‘j’ or just a random arc.. 58

Figure 2.24: Using the colour matching algorithm to break the Gimpy CAPTCHA challenge. As the examples showing, using distinct colour for the text and background noise would make it easy for the OCR to filter the colours.. 60

Figure 2.25: Example of the BotBlock CAPTCHA. Using same colour for the test and background noise would make the CAPTCHA difficult to recognise for the human users. 61

Figure 3.1: The graph represents the rapidly decaying function of visual sensory information following the stimulus offset [52]. The sensory information will remain until about 100 to 150 after the stimulus offset and then starts to wipe off from our memory. 64

Figure 3.2: An example of POV: a series of images playing consecutively to create the illusion of movement. 66

Figure 3.3: Two stimuli are presented with a long ISI delay. As shown in the graph, there is no overlap between the two signals. This means that visual sensory information from the first stimulus is wiped completely before the presentation of the second stimulus. . 67

Figure 3.4: Two stimuli are presented with a very short ISI delay, and thus, there will be some areas where the two signals overlap each other. The overlapping areas will have some information from stimulus-1 and some from stimulus-2. 68

Figure 3.5: The procedure of visual information integration using Trans-saccadic memory. Visual information will perceive during each fixation in our iconic memory. Each saccade will cause our sophisticated visual system to superimpose all the previous perceived visual information with the new visual information [49]. 70

Figure 4.1: Proposed CAPTCHA model development plan (version 1). Different steps needs to be taken in order to generate and play consecutive VICAP frames for the user. 75

Figure 4.2: The image on the left shows a Bitmap image while the image on the right shows a True Type font, which is clearer and smoother around the edges and corners. 77

Figure 4.3: Example of a Bitmap image at various levels of magnification. 77

Figure 4.4: Overview of the CGA. 78

Figure 4.5: The CGA and font drop-down list menu. The user will have a choice of using desirable font type. 79

Figure 4.6: Screen shot of the CGA in its operational phase. 80

Figure 4.7: Randomly generated string of 'AC45R,' converted to a binary format and printed at location (X, Y) against a white background. 81

Figure 4.8: An example of a complete object corresponding to the character 'O' prior to the application of sampling. 82

Figure 4.9: Object 'O' is sampled at 50%. 83

Figure 4.10: Sampling of object 'O' at the same sampling rate of 50%, but with a different pixel combination. 83

Figure 4.11: Adding background noise to the sampled object $\boldsymbol{X}$ at the percentage of $\boldsymbol{n}$%. 84

Figure 4.12: The flowchart represents the process of generating a sampled object and injecting background noise to the VICAP image. 87

Figure 4.13: Steps involved in generating a single VICAP frame. 88

Figure 4.14: Example of 10 different output frames generated and rendered by the CGA with a background noise of 15% and an OSR of 25%. 89

Figure 4.15: General overview of the CTA application. As the picture shows, the application is made of two sections. By clicking on the “Load Frames” button, the process of integration will start. 96

Figure 4.16: CTA application in operational mode. Two graphical representation of output images are presenting, 8-bit/pixel grayscale and 1-bit/pixel mono colour format. 97

Figure 4.17: The procedure of generating VICAP frames using the CGA and feeding them into the CTA to be superimposed. The output results will be an 8-bit/pixel greyscale image as well as a 1-bit/pixel mono-colour image. 97

Figure 4.18: The 256 different shades in the greyscale format image. 98

Figure 4.19: 8-bit conversion according to the MSB and the LSB. 98

Figure 4.20: The summation process of binary values for 5 different frames. As the graph shows, the “Sum” list has all the average value from 0 to 5. Where 0 represent white and 5 represent black colour. All other value in between will represent different shades of grey accordingly. 99

Figure 4.21: The final superimposition output results per 8-bit/pixel in greyscale format. 101

Figure 4.22: A representation of an output rendered image using 8-bit/pixel greyscale format on the left side, and 1-bit/pixel black and white format on the right side. 102

Figure 4.23: A representation of the 256 binary bits divided into different shades in the greyscale format. 103

Figure 4.24: A representation of the 256 binary bits divided into 10 shades of greyscale. 105

Figure 4.25: Comparison of different object sampling rates-OSR (horizontal rows) vs. backgrounds noise ratio-BNR (vertical columns) and the output results in terms of computer recognition success rate (in percentage). 108

Figure 4.26: The process of integrating VICAP into a third-party web server. The requested information will be sent to the CAPTCHA server using secure communication channel. The CAPTCHA server will authenticate the response and grand or decline the access using same communication channel. 109

Figure 4.27: Screen shot of the VICAP challenge integrated into an online application form. 110

Figure 4.28: User has given the correct answer to the CAPTCHA challenge and the CAPTCHA has been accepted. 111

Figure 4.29: An incorrect response has been provided by the user and the CAPTCHA challenge is rejected, denying the user access. 112

Figure 4.30: Final integrated output result from superimposing 10 CAPTCHA frames shown as an 8-bit/pixel greyscale image. A string of 'AE34M' can be clearly retrieved by OCR recognition software...113

Figure 4.31: Various stages involved in breaking VICAP v.1. Stage-1: Superimposition. Stage-2: Binarisation. Stage-3: Background noise removal. Stage-4: Dot-removal and Segmentation. Stage-5: Using OCR and deciphering the correct answer....................114

Figure 5.1: An example of 10 individual CAPTCHA frames generated and rendered by the VICAP-Generator Application with an OSR of 25%, a BNR of 15%, and an ORO of 20%. 118

Figure 5.2: The final integrated output result from superimposing 10 VICAP frames, shown as an 8-bit/pixel greyscale image. No useful information can be retrieved in the second scenario. 119

Figure 5.3: The final integrated output result from superimposing 10 VICAP frames, shown as a 1-bit/pixel black and white image. No useful information can be retrieved in the second scenario. 119

Figure 5.4: VICAP single frame containing random pixels.. 120

Figure 5.5: A screenshot of the GSA-CAPTCHA Breaker showing a 0% recognition success rate for individual frames. 121

Figure 5.6: Uniform presentation of pixels. By presenting all the frames in the uniform sequence it will be easy to predict the missing frame. 123

Figure 5.7: By presenting the pixels in a random sequence, it would be not be possible for the trace machine to predict the behaviour of the pixels by comparing the preceding and succeeding frames. 123

Figure 5.8: Superimposition rendered output image with ORO = 10%....................... 126

Figure 5.9: Superimposition rendered output image with ORO = 30%....................... 126

Figure 5.10: Superimposition rendered output image with ORO = 50%..................... 126

Figure 5.11: Superimposition rendered output image with ORO = 70%..................... 126

Figure 5.12: Superimposition rendered output image with ORO = 90%..................... 126

Figure 5.13: The figure represents three different phases involved in testing our proposed VICAP model. The CAPTCHA images will be produced in phase-1. Then they will superimposed and integrated using CTA application in phase-2. Finally the superimposed output image will be used for recognition software in phase-3. 127

Figure 5.14: A screenshot of the GCB application. VICAP is recognised by the GCB application when the ORO = 90%, with a recognition success rate of 100%.............. 128

Figure 5.15: A screenshot of the GCB application. The GCB application is not able to retrieve any information from the object when the ORO = 20%. In this case, the recognition success rate is 0%... 129

Figure 5.16: A screenshot of the CAPTCHA evaluation website................................139

Figure 5.17: A comparison of CRSR vs. HRSR per different ORO values. As the graph shows, human recognition stands above computer recognition success rate. The ideal situation is measured at 40% where HRSR is at 100% but CRSR is at 50% for a multi-frame and 0% for single frame...143

Figure 5.18: A partial screenshot of a questionnaire designed for our CAPTCHA user experience programme...146

Figure 5.19: The CUEP homepage view...147

Figure 5.20: The participant is presented with the VICAP model..............................148

Figure 5.21: Gender distribution of participants..149

Figure 5.22: Age distribution of participants 150

Figure 5.23: Number of participants who wear glasses. 150

Figure 5.24: Types of monitor used by participants. 151

Figure 5.25: The number of attempts required by participants in order to provide a correct response to the CAPTCHA challenge..152

Figure 5.26: The different time intervals of participants to solve the CAPTCHA test. 153

Figure 5.27: The level of willingness of participants to take part in future challenges.153

Figure 5.28: The level of difficulty in terms of human character recognition.............154

Figure 5.29: The ambiguity level of the characters as experienced by the participants. 155

Figure 5.30: The length of the characters in each CAPTCHA challenge. 155

Figure 5.31: User experience in terms of the size of the characters in each CAPTCHA challenge. 156

Figure 5.32: The distribution of English speaking participants. 157

Figure 5.33: The user experience in terms of difficulty level of the CAPTCHA application. 158

List of Tables

Table 2.1: A brief comparison of different criteria such as Cost, Set-up time, Speed, and Success rate for manual and automatic attacks. 53

Table 2.2: Categorising the CAPTCHAs in terms of Difficulty, Security and Usability, Time to Solve, Setting-up Time, Popularity, and Use of The Human Brain. 53

Table 2.3: Different classifications of CAPTCHAs with their relevant examples and brief explanations of each model. 54

Table 2.4: CAPTCHA usability issues.. 57

Table 4.1: The generation of 10 VICAP frames with a 30% OSR and 15% background noise, followed by the final output rendered images in 8-bit/pixel and 1-bit/pixel formats. 91

Table 4.2: The generation of 10 CAPTCHA frames with the same ratio for the OSR and the background noise, both 15%, followed by the final rendered output images in 8-bit/pixel & 1-bit/pixel formats. 92

Table 4.3: The generation of 10 CAPTCHA frames with an OSR of 15%, and a background noise of 30%, followed by the final output rendered images in 8-bit/pixel & 1-bit/pixel formats. 94

Table 4.4: Representation of the pixel binary values for 5 frames and the summation value of all the pixels. 100

Table 4.5: The generation of 5 CAPTCHA frames with an OSR of 25% and a BNR of 15%, followed by the superimposed output images in an 8-bit/pixel greyscale format and a 1-bit/pixel black and white image. 104

Table 4.6: The generation of 10 CAPTCHA frames with an OSR of 25% and a BNR of 15%, followed by the superimposed output images in 8-bit/pixel greyscale format and 1-bit/pixel black and white image. 105

Table 5.1: Comparison of CRSRs versus the ORO parameter. 130

Table 5.2: The change in CRSRs for an entire CAPTCHA string when varying the ORO parameters. 131

Table 5.3: Change in the CRSRs for partial CAPTCHA character recognition when varying the ORO parameters. 131

Table 5.4: Partial recognition success rate calculation methodology. 133

Table 5.5: Recognition output results for different CAPTCHA decoders. 136

Table 5.6: A comparison of different ORO frame rate parameters and the effect these have on the HRSR. 140

Table 5.7: Changes in the HRSR for the entire CAPTCHA string when varying the ORO parameters. 141

Table 5.8: Changes in the human recognition success rate for partial CAPTCHA character recognition when varying the ORO parameters. 142

Table 5.9: The average value of the CRSR for 220 VICAP images analysed individually by an OCR programme. 144

Table 5.10: Best and worst-case scenarios for recognition success rates for human users and computer programmes when varying ORO levels. 160

Listings

Listing 2.1: Horizontal line removal algorithm..45

Listing 2.2: Vertical line removal algorithm..46

Listing 4.1: Algorithm used to sample the object according to the specified threshold value. 84

Listing 4.2: Algorithm used to add background noise to the sampled object according to the defined threshold value. 85

Listing 4.3: Algorithm to convert 5 different shades of grey according to the summation of 5-pixel values. 101

Chapter 1

Introduction

This section aims to produce a comprehensive research review and analysis of current state-of-the-art Humans Interactive Proofs (HIPs) and their new definition of CAPTCHAs (Completely Automated Public Turing Test to tell Computers and Humans Apart), which are critical components in computer and web security in order to distinguish between human users and automated computer programmes.

1.1 Human Interactive Proofs (HIPs) and CAPTCHAs

As the number of online users is growing fast, the number of online threats is growing just as rapidly. These days, the number of computer attacks (including automated computer programmes) that are targeting websites and online companies are becoming commonplace. There are numerous different kinds of threats to computer systems that can act as human users, and attack systems by providing similar identification information to a human user in order to register themselves to a system in exactly the same way as a human user would do [1]. For this reason, there is high demand for a mechanism that can distinguish between real human users and computer automated programmes (also known as bots). The process of User Identification and the reasons for having a robust online security system in place have been discussed, so it is now prudent to examine the types of CAPTCHAs currently being used.

HIPs are a set of challenge or response protocols that have been designed in the form of a challenge or a test that can be presented to a user to tell human users and computer automated programmes apart [2]. HIPs are designed to be easy for a human user to solve, whilst being too hard for almost any automated computer programme to break. HIPs can also be used to distinguish between a computer bot and a human user (or group of users) from another group of users, as well as distinguish between a user and another user on the specific IT related task [3]. As mentioned before, HIPs can be presented in different forms of visual challenge such as text or graphical images, or it can be in the form of non-visual challenges such as audio tests. HIPs should also be designed in such a way so that no current computer programme could break them, whilst at the same time be easy for humans to solve without discouraging them from using the service again. For instance, as seen in Figure 1.1, a combination of distorted letters and numbers with some background noise would be presented to a user. The user would then need to read and recognise all the distorted letters and numbers correctly and retype all the correspondent ASCII codes for every single letter and number, which in this case would be ‘D98LDGNV.’

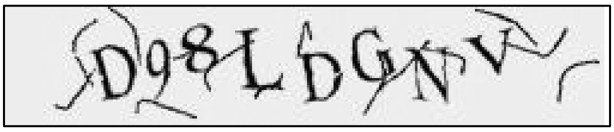


Figure 1.1: Example of Text-based HIPs. It should be noted that letters in this CAPTCHA are deformed and the lines that have been added create problems for an OCR system. [3]

HIPs are designed to keep a system safe from being attacked or hacked by potentially automated computer programmes. CAPTCHA is a class of HIP whose main objective is to distinguish between human users and automated computer programmes. CAPTCHA was first introduced in 2000 by Luis von Ahn and his team at the Carnegie Mellon University [4]. According to its definition, a CAPTCHA should have at least two main conditions to be recognised as a CAPTCHA. Firstly, the test should be easy for a human user to solve, whilst it must be too hard (or almost impossible) for a computer programme to break. Secondly, the test should be able to be automatically generated by the system and any code or algorithm used to create the challenge should also be publicly available [5]. This means that even if an attacker has got the script code being used to generate the CAPTCHA challenge, the attacker will still not be able to break the CAPTCHA. Figure 1.2 shows a different type of CAPTCHA called the ReCAPTCHA [6], which is a different type of CAPTCHA technology that will be explained in more detail in Chapter 2 of this study.

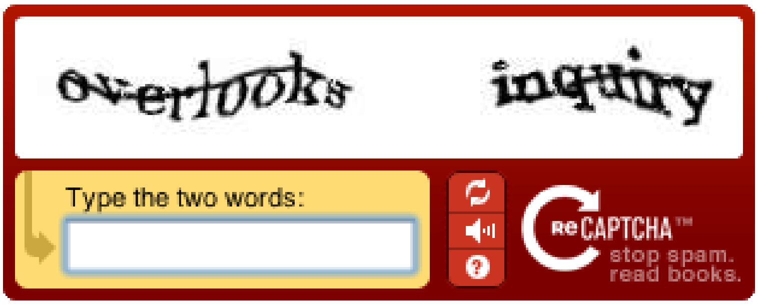


Figure 1.2: An example of ReCAPTCHA by google. The image shows two distorted words and the user is required to decipher and recognise [6]

CAPTCHAs are being used widely as security and anti-scamming measures for different websites and online services who need to tell human users and automated computer programmes apart. CAPTCHAs can also be used for many different applications such as free email providers like Yahoo! and Gmail, and online ticket sellers like Ticket Master, as well as online polling stations and chatrooms.

The idea of having a CAPTCHA was first introduced to avoid a system being hacked by an automated computer programme and to make sure that only human users can access a website. For instance, according to the results of a study which was done by MSN Hotmail when they deployed their first HIP challenge on their website, Hotmail registration dropped by almost 19% without having any impact on their customer support inquiry [3]. But according to this research, provided in source [3], it can be observed that this drop in registration was attributed to the drop in automated scripts that were being used to create fake email accounts, which were then used for the purpose of scamming. However, attackers are getting more sophisticated every day and they are also increasingly able to solve more and more CAPTCHAs. Thus, CAPTCHA challenges also need to improve rapidly as they are still vulnerable to ever increasing automated scripting attacks. For that reason, it is vital to conduct research into this area as a means to recognise possible security gaps and fill these by improving the current CAPTCHA models and technologies.

Figure 1.3 is a very simple but clear sketch of the function of HIPs according to the difficulty level for human users and computer programmes [3]. As it can be observed from the graph, the function of HIPs has been divided into three different regions. The first region on the left-hand side represents the area by which a test is easily understandable and solvable by computer recognition programmes, meaning that the test is very easily broken by bots. The middle region is the ‘ideal region’ by which a test is unsolvable to most computer programmes, yet it is solvable to almost all the human users. The last part of the graph on the right-hand side is the region by which the test is just too hard for any human user to solve.

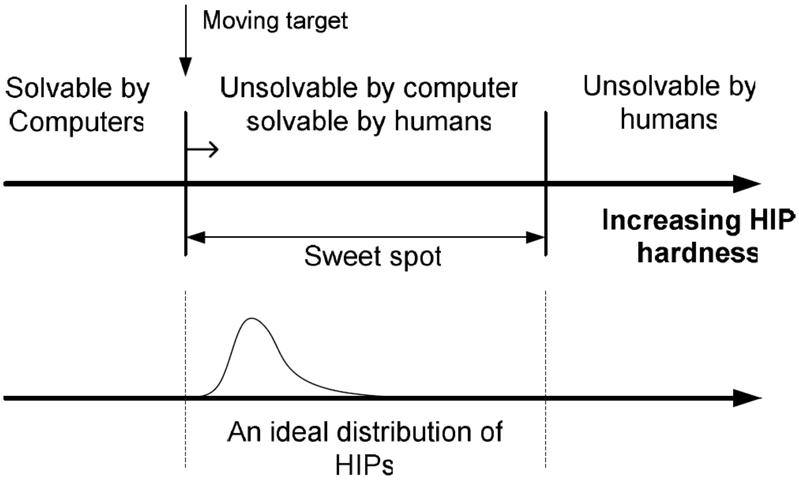


Figure 1.3: Changing of the function of HIPs according to its difficulty level for computer programmes and human users [3]

As it has been shown in Figure 1.3, the first generation of HIPs was not sophisticated enough, and even very ordinary computer recognition software could break them easily. As the difficulty level of HIPs increases, the possibility of computer programmes breaking the test also decreases. As it can be observed from the graph, as the difficulty level sharply increases, so does the slope of the graph, until it reaches what could be called the ‘peak point,’ which is our ideal level. That peak point represents the ideal balance between a HIP test being easily solvable for human users, but not bots (marked as the ‘sweet spot’). As the difficulty level of the test increases further in Figure 1.4, so does the possibility of the test being too difficult for a human user. As the difficulty level of the test increases, the slope of the graph also drops sharply until it reaches 0, which would render the test pointless because it would mean that it would be just too hard for almost all human users. Consequently, there is always a trade-off between the difficulty level of CAPTCHAs and their user-friendliness. For all these reasons, it is critical to creating a CAPTCHA model that is just as sophisticated as the computer bots that are attempting to break them, yet also remain user-friendly [3].

The concept of distinguishing between computer programmes and human users dates back to 1950 when the original Turing Test was introduced by Alan Turing [7]. This test involved asking a human (or a judge) to distinguish if another player is a human or a computer programme by asking a series of questions. CAPTCHAs work similarly to the Turing Test as their sole purpose is to distinguish between human users and computer programmes. Yet, as explained previously, there is a key difference between the Turing Test and the CAPTCHA challenge, namely that in the Turing Test the judge is a human, while, in the CAPTCHA test the judge is a computer programme. For this reason, in some publications such as Kumar Chellapilla's resource entitled *'Designing Human-Friendly Human Interaction Proofs (HIPs),* a CAPTCHA is labelled as a 'REVERSE Turing test' [3]. The idea of this research project is to introduce a novel CAPTCHA model based on human's psychophysics called "Persistence of Vision" or POV. Since this is a unique ability of human's visual system to remember and superimpose all the seen frames using Iconic Memory (IM). Therefore, it is believed that no current OCR programmes would be able to break this novel CAPTCHA model. More information about this technique and its procedure will be provided in the next chapters.

1.2 Research Objectives

According to cybersecurity researchers, there is a huge concern on the security of the current CAPTCHA models. As results of that, IT experts announced that newly developed CAPTCHA breaker software called "DeCaptcha" can break audio CAPTCHAs up to 89% success rate. Also as the results of research showing, current Text-based CAPTCHA models have been broken with high success rate. For instance, eBay CAPTCHA was broken at 82% and Microsoft CAPTCHA was broken with 42% success rate [8]. The overall aim of this PhD research project is to develop a novel CAPTCHA model that can distinguish between human users and automated computer programmes with the purpose of improving the general security of online activities. The proposed CAPTCHA model should be easy for humans to solve, taking minimal effort and time, whilst remaining too difficult for current computer recognition programmes to break. To achieve this goal, several objectives have been formulated:

1- To carry out background research on state-of-the-art CAPTCHAs and HIPs in order to collect and compare current CAPTCHA model specifications qualitatively and quantitatively. Also, to investigate and analyse any possible security gaps and weaknesses associated with each type of CAPTCHA model that may increase the risk of bot attacks. In addition, a brief literature review on Persistence of Vision (POV) and optical illusions will be presented in order to better understand how the HVS works in terms of psychophysics.

2- To develop and propose a novel CAPTCHA model based on POV that can only be meaningful to human users and has absolutely no meaning for computer recognition programmes. This novel CAPTCHA model, entitled VICAP has been designed in such a way so as to be robust against current Optical Character Recognition (OCR) software. However, at the same time, it has been designed to be easy for humans to recognise and pass, according to the definition of CAPTCHA.

3- To test and evaluate the proposed CAPTCHA model based on POV against different CAPTCHA attacks, as well as the most sophisticated OCR software in order to measure its robustness. The level of robustness can be measured quantitatively by producing a table that represents the areas where the proposed model was vulnerable to the attacks.

4- To test and evaluate the proposed CAPTCHA model on real human users using qualitative and quantitative methods. A comprehensive questionnaire has been designed for this purpose, including a range of different questions. Every question was designed to measure an aspect of usability both quantitatively *and* qualitatively. A website called the CAPTCHA User Experience Programme has been developed in order to demonstrate and compare our proposed CAPTCHA model with one of the most famous Google CAPTCHA models called ReCAPTCHA.

1.3 Contributions

Our main contributions to this research project can be listed as follows:

- The concept of visual psychophysics is used in this project to design a novel CAPTCHA model, which would only be understandable for human users and not current computer recognition programmes.
- POV has been applied to the new CAPTCHA model to superimpose and integrate all the CAPTCHA images to form an object using the human brain.
- The Recognition Improvement Level (RIL) for human users has been seen to have a 50% increase in recognition success rates compared to current computer recognition programmes based on multi-frames scenarios.
- The new CAPTCHA's ability to be unrecognisable to current OCR programmes has increased by over 99% based on single-frame scenarios compared to the current text-based CAPTCHA models.
- In terms of usability and performance, our proposed VICAP model has improved by 92% in terms of time to solve, and similarly 65% in terms of difficulty level. Furthermore, as the user satisfaction results confirm, we have achieved a 35% improvement in terms of clarity and ambiguity level of the CAPTCHA characters.

1.4 Outline of Thesis

This PhD thesis has been structured as follows:

Chapter 1 begins by introducing what is meant by User Identification, giving a brief overview of HIPs, and also the reasons why online service providers and websites try to recognise each user as genuine. To increase the level of security of online service providers, there is a need to have a mechanism to separate machines from humans. Additionally, the concept of having a test that human users can pass but computer programmes cannot pass will be introduced. These tests are also known as CAPTCHAs.

Chapter 2 will present a literature review of currently existing CAPTCHAs and also give an overview of the highly influential Turing Test (TT) and Imitation Game (IG), highly relevant to the concept of CAPTCHAs. In the rest of the chapter, the definition of

CAPTCHA has been explained as well as the difference between the CAPTCHA method and the TT. As the use of CAPTCHAs is to distinguish between human users and bots, the different threats and attacks that can harm the security of human users has also been discussed in this chapter. Furthermore, the basic categorisation of current CAPTCHA per OCR-based and non-OCR-based programmes has also been presented. Most OCR-based CAPTCHAs can be broken and hacked using OCR techniques because they are based on text and characters. Non-OCR-based CAPTCHAs are more robust against computer attackers and are more user-friendly, but are also more complicated and more expensive to make. Chapter 2 will end by comparing the OCR-based and non-OCR-based CAPTCHA in terms of usability, time to solve, robustness, and cost.

Chapter 3 will give a brief overview on human's psychophysics and precisely Persistence of Vision. Also, the proposed temporal integration and Trans-Saccadic Integration techniques will be discussed in this chapter. The concept of Persistence of Vision (POV) is the reason that humans can see the world continually after each blink of the eye. POV is a well-known phenomenon in the movie industry as it means a movie can be watched smoothly without being interrupted by flashing images. The human memory system is made up of different parts that have a bi-directional correlation together. It is thought that Iconic Memory (IM) is the main cause of POV as it helps the HVS retain all perceived visual information for a fraction of a second. Our new proposed model uses this ability to create the final image of a hidden character in our visual system. Since computers do not possess this ability, this model cannot be recognised by current computer recognition programmes. Chapter 3 finishes by presenting our novel mathematical model using a trans-saccadic visual integration technique based on single-stage scenarios (SSS) and multi-stage scenarios (MSS).

Chapter 4 will introduce our proposed CAPTCHA version.1 model called "VICAP-v.1". The various stages that are involved in rendering and producing this VICAP-v.1 model have been discussed, such as binarization, the Object Sampling Rate (OSR), and the adding of background noise. The CAPTCHA-Generator Application (CGA) is a state-of-the-art application developed for this research. The idea of the CGA is to render and generate VICAP frames and to play them at a very high speed for the end user in order to be solved and recognised. A CAPTCHA-Test Application (CTA) was also developed to test the proposed CAPTCHA model, which works in a similar way to the human eye in order to superimpose and integrate all of the perceived CAPTCHA frames. A series of experiments have been conducted in laboratory conditions to find the best possible combination for the OSR and the Background Noise Rate (BNR) in terms of the CRSR. Comprehensive experimental superimposed simulation results have been produced based on 5 and 10 frame sequences in order to find the impact of the total number of frames on the recognition output results. Integrating the proposed CAPTCHA model into websites has also been discussed in this chapter. This chapter concludes our study and evaluation results of the proposed CAPTCHA model, and also address the possible weaknesses of the model in terms of its security.

Chapter 5 will start by introducing a new parameter called Original to Random Output data (ORO). In the previous chapter, the lack of security of the proposed CAPTCHA model was addressed; thus, an additional ORO parameter has been introduced in order to increase the robustness of the VICAP model v.1. This latest version labelled VICAP v.2, has been analysed against most current CAPTCHA attacks and the output results have been presented. Moreover, the improved version of the VICAP has been tested in laboratory conditions against the most sophisticated computer recognition software, and the output results are presented in this chapter. Lastly, the VICAP v.2 model has been tested on diverse groups of users to measure its usability compared to the current Google CAPTCHA (ReCAPTCHA).

Chapter 6 will finish by concluding the ideas of the whole dissertation. Even though the fundamentals of CAPTCHAs and HIPs have been discussed in this thesis, the sophistication of CAPTCHA models and bot attacks are still rapidly increasing. The new CAPTCHA model is proposed in this research project is proven to be significantly robust against current famous OCR systems with high human recognition success rate and very low computer recognition success rate.

Chapter 2

State-of-the-art in Human Interactive Proofs and CAPTCHAs

2.1 Introduction

In the Introduction, a brief definition of HIPs and CAPTCHAs were provided. As already outlined, the need for CAPTCHAs, which is to increase the security of online services, is rapidly increasing. This chapter will provide more information on CAPTCHAs by giving a brief history of CAPTCHAs and the TT, which is a similar concept to CAPTCHAs. Following this, more information will be given on state-of-the-art CAPTCHAs by introducing some of the applications that could benefit from using them to increase their robustness against different online attacks. To end this examination of CAPTCHAs, some of the most critical threats to online services will also be discussed.

2.2 The Turing Test and Imitation Game

The TT was firstly developed at the Computing Laboratory at Manchester University by Alan Turing in 1950. The purpose of this test was to distinguish between human users and computers by asking series of questions. Turing's idea was based on the question: '*Can machines think?*' [7]. He proposed this idea by developing a game called the Imitation Game (IG). The first version of the IG consisted of three players, where player A is a man, player B is a woman, and player C is an interrogator whose gender is unimportant. Player C will then separate from the others and be isolated in a room with no physical access to the other players. They then start to communicate with each other in writing. The objective of the game is for player C, the interrogator, to recognise which player is a woman, while the other two players are supposed to convince the interrogator that he or she is the woman, as it has been showing in Figure 2.1.

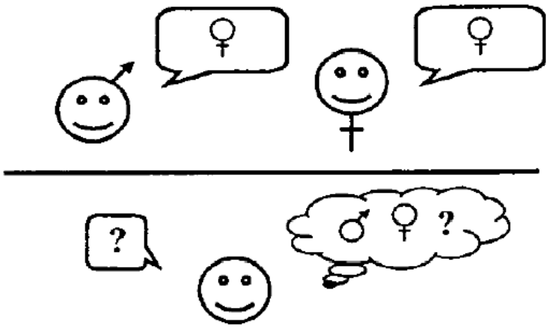


Figure 2.1: First version of the IG. The role of the interrogator is to identify which one of the two players is a woman [9].

2.2.1 Standard Interpretation

To answer Turing’s question: *‘Can machines think?’* player A is replaced by a computer and player C does not try to determine which one is male or female, but instead tries to work out which one is the computer and which one is the human. The TT does not aim to understand how intelligent the computer is, but can only examine whether the computer behaves like a human being or not. Figure 2.2 shows the second version of the IG, also known as the TT.

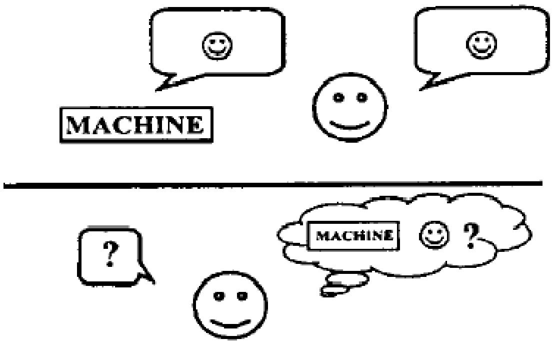


Figure 2.2: Second version of IG known as the TT. As it is shown in this example, the role integrator is to identify which of the two players is a machine and which one is human. [9]

In this example, the interrogator (player C) will pose some different questions to both players and will make a judgment based on their responses. The questions can be anything, such as: ‘How is the weather today?’ or ‘What is your favourite movie?’

Turing's idea was to test how intelligent computers can be and how much they can imitate a human's identity. Since there are differences in the ways humans think and computer programmes solve problems, there are often tell-tale signs that help the interrogator recognise which player is human and which one is a machine.

A computer can fail the TT due to two key weaknesses:

1. **Some human behaviours are unintelligent and cannot be imitated.** During the TT, a human would act exactly as a human would, regardless of whether the behaviour is intelligent or not. For instance, typical human behaviour stemming from human emotion and ability, like susceptibility to insults, lying, or typing mistakes are unique to humans, and computers cannot imitate these unintelligent behaviours. This would cause them to fail the test.
2. **Some intelligent behaviours are not human.** If there is a problem that only a computer programme can solve, by asking such a question, the interrogator could easily recognise whether the other user is a computer programme or a human, which will subsequently cause the test to fail. Figure 2.3 shows clearly that the TT falls into the intersecting area between typical human behaviour and intelligent behaviour [10].

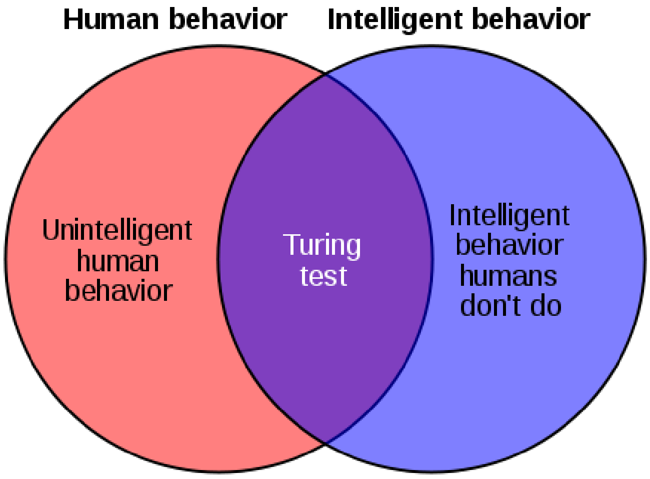
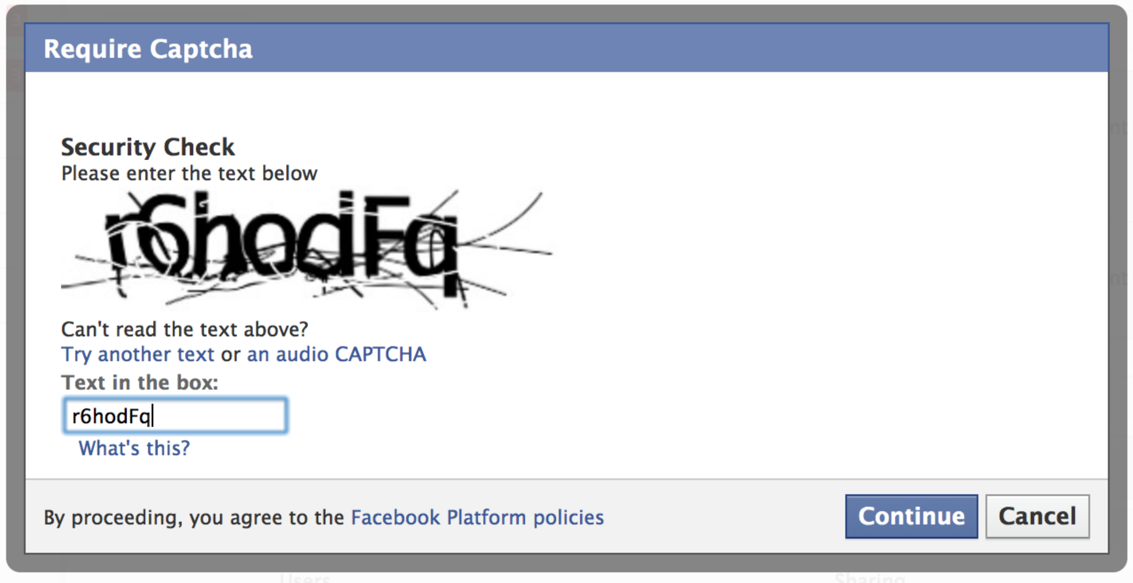

Human behavior
Intelligent behavior
Unintelligent human behavior
Turing test
Intelligent behavior humans don't do

Figure 2.3: TT falls into the intersection area between typical human behaviour and a computer's intelligent behaviour [10].

2.3 CAPTCHA Concept

Nowadays, most of us have experienced typing out distorted character images when using online forms and websites. These images are annoying and time-consuming to recognise, and sometimes they even prove unreadable. As has been stated, these distorted images, known as CAPTCHAs, are used to protect websites and online services from possible bot attacks. The key role of CAPTCHAs is to distinguish between human users and bots. An example of the current text-based CAPTCHA model used by Facebook is displayed in Figure 2.4.



Require Captcha

Security Check

Please enter the text below

Can't read the text above?

Try another text or an audio CAPTCHA

Text in the box:

r6hodFq

What's this?

By proceeding, you agree to the Facebook Platform policies

Figure 2.4: An example of Facebook CAPTCHA. The user is required to recognise all the letters displayed in the distorted image in order to access the website [11].

CAPTCHA was first introduced by Luis von Ahn, Manuel Blum, Nicholas J. Hopper, and John Langford in 2000 in Carnegie Mellon University [4]. In some publications, it has also been called the Reverse Turing Test [3]. It is labelled as ‘Completely Automated’ because a computer system generates and creates the CAPTCHA test completely automatically with no need for any human input. Also, it is also labelled as ‘Public’ because, by definition, the source code of a CAPTCHA should be publicly available. Saying this, the programme should still be designed in such a way that, even with the source code, no computer programme would be able to break the algorithm behind CAPTCHA or hack into the system behind it. The CAPTCHA system was designed to defeat OCR systems in order to make websites secure. According to the aforementioned resource [12] written by Luis von Ahn, a CAPTCHA is a programme that can generate a test that humans can pass, but computer programmes cannot pass. Furthermore, according to that resource, any programme that can solve a CAPTCHA could also solve an Artificial Intelligence (AI) problem.

2.4 How Does a CAPTCHA Work?

As shown in Figure 2.5, a CAPTCHA is generated by an online server and a user would respond to a challenge/response environment. When a client requests a service from an online entity, he or she firstly sends a message to the web server. Depending on the nature of the online service, the web server will then decide whether to allow the user access to the requested resource or else authenticates the user before allowing them access. Where any requested resources are protected, the web server will invoke the CAPTCHA generator application to create a new challenge, which is then sent to the user via the same communication channel. The user is then required to solve the challenge and provide the answer to the request to prove that they are a human user. The CAPTCHA generator application is made up of two separate elements. The first element is the CAPTCHA image database, which consists of all the information that a CAPTCHA application is using to measure the level of accuracy (in other words, whether the answer is correct). The second is called the CAPTCHA verification element, which works as a judge. If the input data from the user matches the CAPTCHA image stored in the database, then the user will be authenticated and will gain access to the online services. However, if the user response from the client side does not match the database information on the server side, then the CAPTCHA generator application will deny access, and the user should try this process again until he or she gives the correct response.

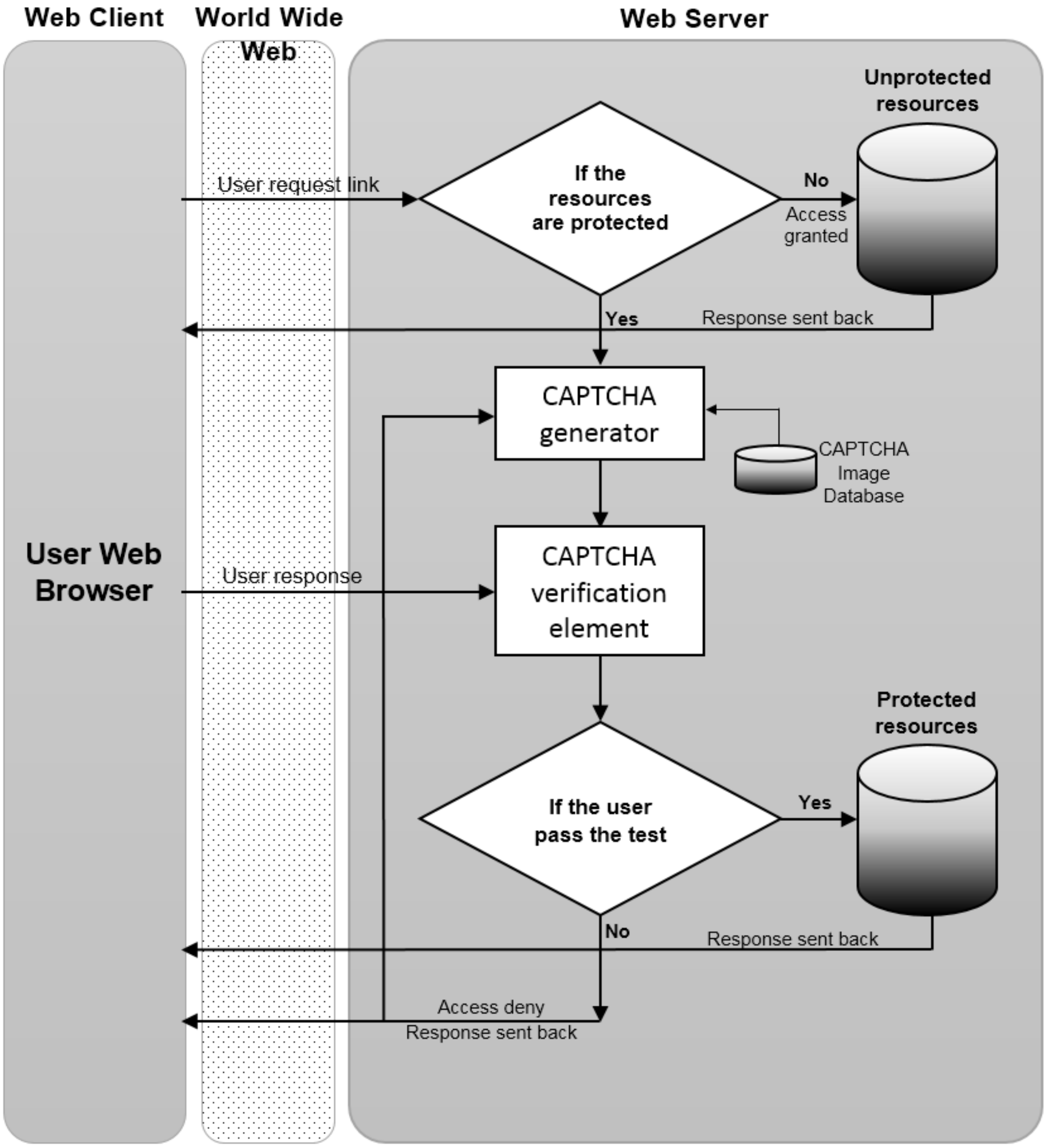


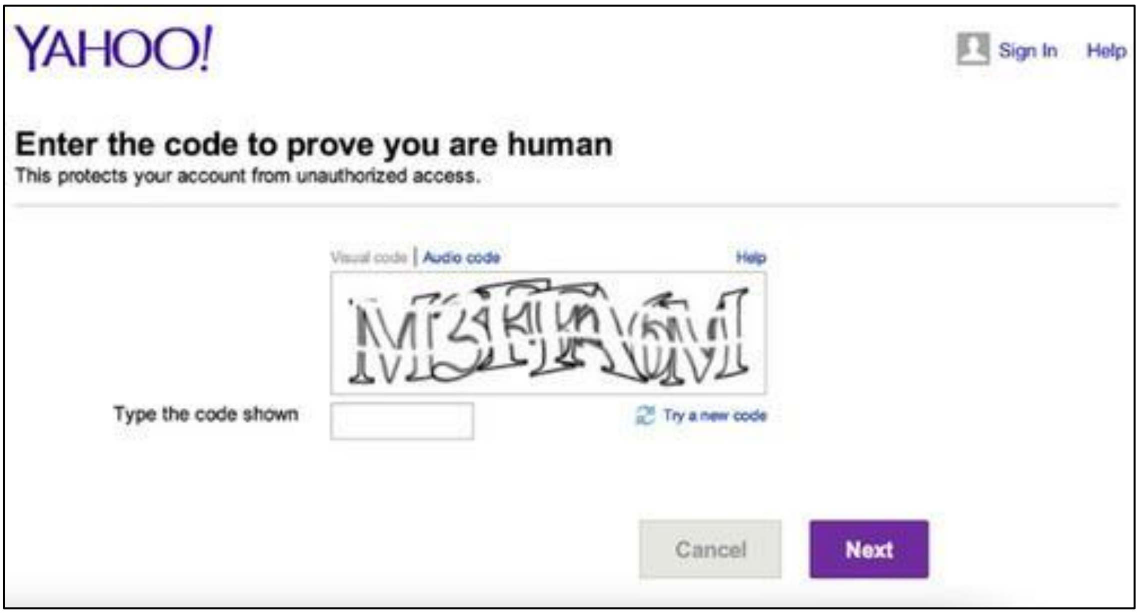
Figure 2.5: General framework for the CAPTCHA authentication process.

2.5 CAPTCHA Applications

There are different applications that could use CAPTCHAs to increase the security and robustness of their systems. From all the different applications that currently use CAPTCHAs to prevent bot attacks, five core applications can be reviewed as follows [13]:

- **Online Polls** - In online polling systems, there are certain websites that users need to enter to cast their vote. To avoid having two votes from a single user, the online polls usually keep the IP address of that computer and will not allow that IP address to be reused again to cast another vote. The first problem appeared in November 1999 when a website called *slashdot.com* started asking graduate students to vote for the best graduate school in the United States [5]. The next day, students of Carnegie Mellon University found a way to trick the system into writing a programme that could generate different IP addresses randomly and vote thousands of times for CMU.

- **Free Email Services** - Free email account services such as Yahoo!, Hotmail, and Microsoft also started suffering from bot attacks. As has been stated, bots are computer software that can register themselves to a server in exactly the same way as a human can, meaning they can sign up for thousands of email accounts in under a minute. Thus, it is vital for email service providers to distinguish between human users and computer automated programmes. Yahoo! was the first email service provider to use CAPTCHAs, an example of which can be seen in Figure 2.6.



Enter the code to prove you are human

This protects your account from unauthorized access.

Figure 2.6: Example of a Text-based CAPTCHA used by Yahoo! It can be noticed that the text is reformed in order to make it difficult for the OCR programmes to recognise [14].

- **Search Engine Bots** - Visiting a website many times will increase its index and it will be hit by search engines more often. In other words, searching a website will rank it to the top of the queue in search engines. Search engine bots can enter any website the same as a human can. As some websites do not want to be indexed by search engine bots, the job of CAPTCHAs is to make sure that only human users can enter the website, not computer programmes.

- **Worms and Spams** - CAPTCHAs can also be used to make sure there is a human behind every sent email to avoid the sending of worm and spam emails. There are a few companies already working in this area, such as *www.spamarrest.com.* All researchers agree that there is no way to prove a CAPTCHA is a guaranteed solution to prevent attackers. There is also no proof that computer programmes cannot be as sufficient as humans in many respects.
- **Preventing Dictionary Attacks** - CAPTCHAs can also be used to prevent dictionary attacks by giving a series of automated randomly generated answers on the system [13]. This is a very simple way of protecting an account from getting locked by an attacker after they have attempted to guess a password many times. CAPTCHA can also be used to ask a user to solve a CAPTCHA problem after giving the wrong answer a couple of times. In this case, instead of giving a computer attacker the chance to try different random passwords and locking an account, a CAPTCHA can be used to recognise if the user is a human user or an automated computer programme.

2.6 The Basic Categorisation of the CAPTCHAs

CAPTCHAs are available in many different formats depending on their specifications and security functionality. We can classify CAPTCHAs into three main categories: OCR-based, non-OCR-based and non-visual-based [15]. Figure 2.7 is a representation of the different CAPTCHA types based on their classifications, where each class has its own conditions and specifications. Based on previous research, OCR-based (or text-based) CAPTCHAs are the most popular to date. These types of CAPTCHAs are easier in terms of web integration and implementation. In addition, they are more efficient in terms of design and evaluation. The reason these types of CAPTCHAs are labelled as being OCR-based is that the challenge is made up of different distorted characters and text. Also, OCR software would need to be used to break the CAPTCHA.

Non-OCR-based CAPTCHAs utilise multimedia features such as video, images, or pictures. Examples of these types of CAPTCHAs are the Collage CAPTCHA [16], PIX [17], Bongo [17], Asirra [18], and GeoCAPTCHA [19], all of which are based on the recognition of an image or a group of images using some specific criteria. There is another type of CAPTCHA that is not based on visual recognition, known as an audio-based CAPTCHA. The purpose behind audio-based CAPTCHAs was to overcome some of the weaknesses of OCR-based CAPTCHAs. Specifically, in some OCR-based CAPTCHAs the text is too distorted or significantly deformed, and therefore, it is cumbersome for users to read and recognise. In that case, they usually put the audio version of the same text as an alternative way of solving the CAPTCHA. This version of the programme randomly picks a word or a sequence of numbers, then transforms them into a short audio clip, which is mixed with a reasonable level of background noise. It then presents the distorted audio clip to the user, who must then recognise and type out the content of the sound clip [20]. This method is based on the auditory ability of humans to recognise distorted words or numbers with a moderate level of background noise present. It is very difficult for most voice recognition software to distinguish between a central voice and background noise [20].

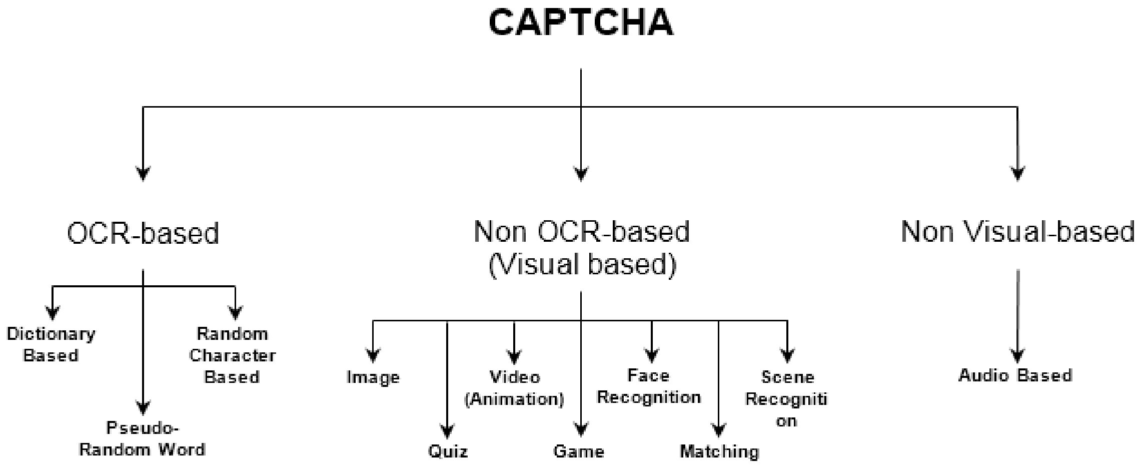


Figure 2.7: The basic categorisation of CAPTCHAs. As the figure shows there are three main categories of CAPTCHA according to their nature with examples of each.

2.7 Security of CAPTCHAs

Robustness and security of CAPTCHAs have been the centre of attention in the online and cybersecurity research fields for many years. As the number of online threats is growing at a high rate, the importance of making a CAPTCHA challenge more secure and robust is becoming even more critical. In this section, some of the most important CAPTCHA attacks will be discussed briefly, which threaten most models. Text-based CAPTCHAs are most vulnerable to attacks compared to any other type of CAPTCHA (such as image-based). Thus, more detailed explanation about the main threats associated with these types of CAPTCHA will be provided. As mentioned earlier, the number of bot attacks to online services is huge. In this section, the most significant threats currently being used to break CAPTCHAs will be discussed.

2.7.1 Automated Character Recognition Software Attacks (OCR)

Since OCR-based CAPTCHAs are the most vulnerable type, this section will focus on key aspects of their security [21]. As shown in [22], most OCR-based CAPTCHAs have been broken with over 92% success rate using different recognition and segmentation techniques. OCR software is electronic software used to convert handwritten documents or typed glyphs and words into digital format. Most OCR recognition programmes use pattern recognition techniques, such as vertical segmentation, to separate each glyph and character. This is done to recognise every single character by comparing all the pixels with pre-stored characters on a database [23].

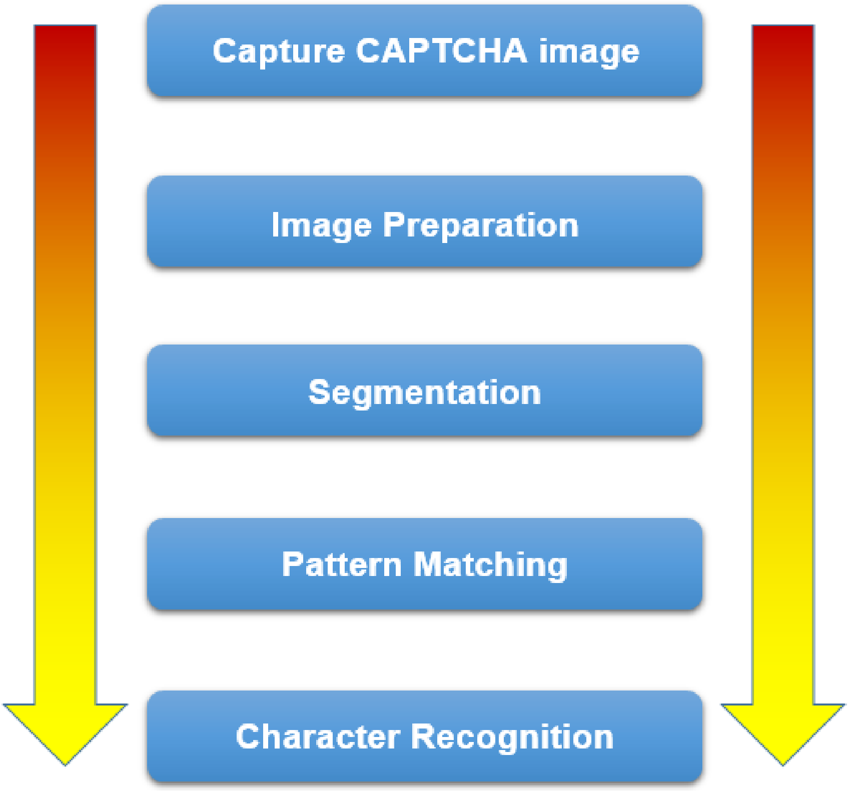


Figure 2.8: The graph shows various stages involved in the OCR recognition process.

OCR recognition programmes rely on different techniques to recognise characters or words. Image preparation or pre-processing techniques are used to convert the image into a greyscale or binary format. The pre-processing phase can also include background noise or line removal. Segmentation is a technique to separate each glyph and character in order to analyze the pixel values for each separately, which are then matched with the characters pre-stored on the OCR database or a dictionary [24]. Recognition (or pattern matching) is the last step that OCR programmes use to match the separated characters to similar characters and glyphs stored on their database (which is dictionary-based). The different processes involved in the recognition of text-based CAPTCHAs are shown in Figure 2.8.

Segmentation is a very important procedure in any OCR recognition programme, and it is needed to separate the characters from each other to make recognition possible. According to [22], there were several numbers of 2D-CAPTCHA models believed to be resistant to segmentation. However, as the outcomes of Jeff Yan and Ahmad Salah El Ahmad have shown, most 2D-CAPTCHAs can be broken using vertical segmentation techniques [25]. In the same resource, there is also another technique called pixel count, which counts the total number of pixels in each chunk (or segment). By comparing the total number of pixels with the character information on its database, it would easily be able to map the corresponding character [26]. As mentioned earlier, to break a text-based CAPTCHA, there are several different procedures involved. A short presentation on the different phases involved in breaking the Ez-Gimpy CAPTCHA [27] model is presented below, from image preparation to character recognition and pattern matching. OCR software will start breaking the CAPTCHA by seizing the CAPTCHA image and carrying out image processing techniques.

1. **Image Preparation:** After seizing the CAPTCHA image, there will be some modifications to it to make it easier for it to be processed and analysed. The purpose of the preparation phase is to remove the background noise from the image to make recognising it easier. The preparation phase consists of the following procedures:

    - **Converting to Greyscale:** Any colourful image consists of three main components (RGB), and the processing of these is very difficult. To make the processing job easier for the computer programme, they first convert the image into a 'greyscale' format to reduce the number of components per pixel. In greyscale mode, there are only 256 intensity values that exist per pixel.

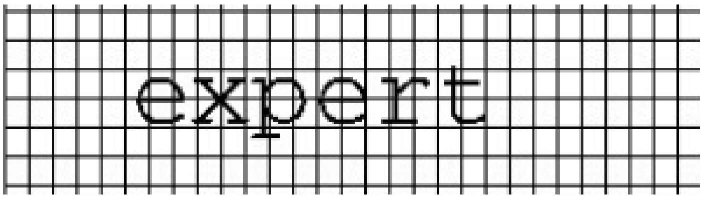


Figure 2.9: Converting the CAPTCHA image into greyscale mode [27].

- **Converting to Binary Image:** Binary images are images that have pixels with only two possible values: 0 or 1. They are normally presented as black and white images where 0 represents black pixels, and 1 (or 255) represents white pixels. In this case, binary images are often used to distinguish between the background noise and the foreground image of a CAPTCHA. This means that any value less than the threshold value is replaced with 0 and any value greater than the threshold is replaced with 1. The background is usually represented by black pixels, and the foreground image represented by white pixels.

- **Removing Background Lines and Noise:** As Gimpy CAPTCHA images are presented on a background of straight lines, it is very important to remove these and any background noise from the CAPTCHA image. The background noise removal phase is divided into two steps, which use different algorithms as follows: *"Horizontal Line Removal Algorithm"*, is to remove horizontal lines from the CAPTCHA background, a pseudo-code has been developed, as shown in Listing 2.1. *"Vertical Line Removal Algorithm"*, is to remove vertical lines from the CAPTCHA image, a pseudo-code has been developed, as shown in Listing 2.2. After applying the above algorithm to the CAPTCHA image, the following result can be obtained, as shown in Figure 2.10.

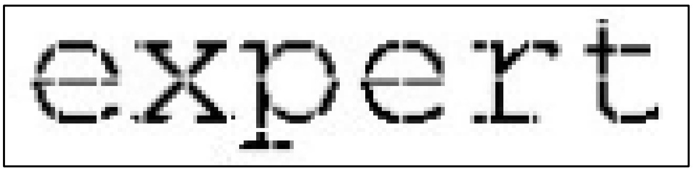
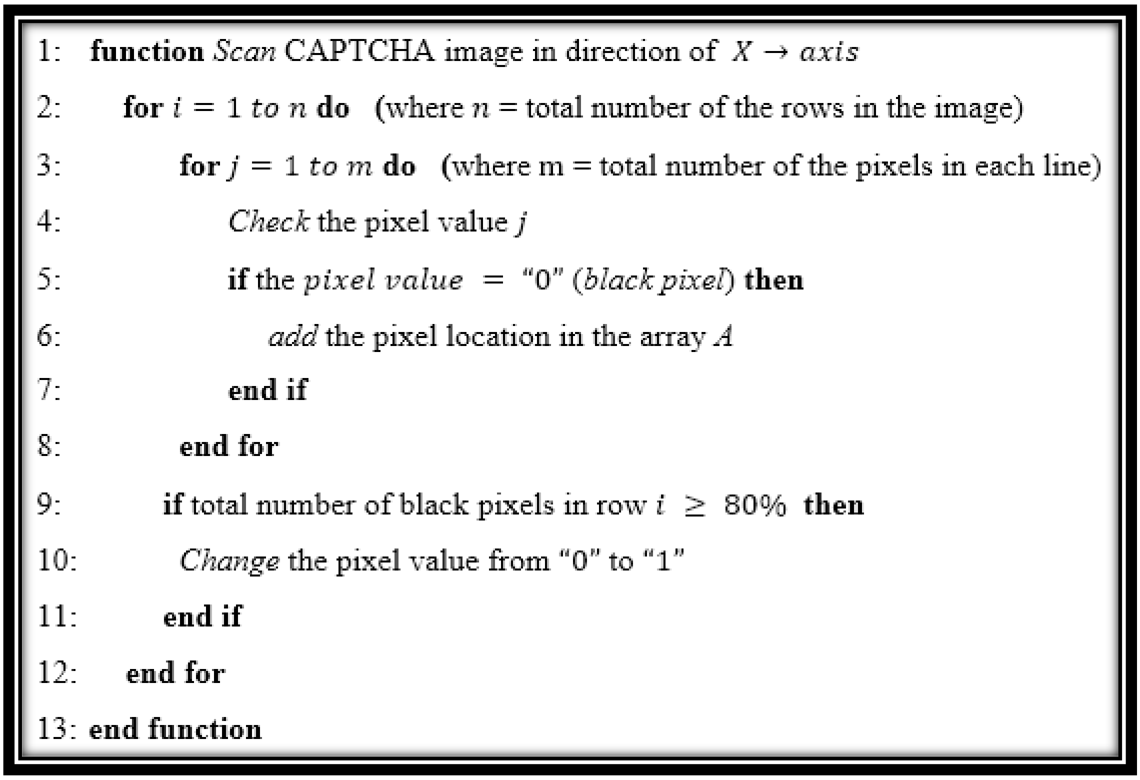
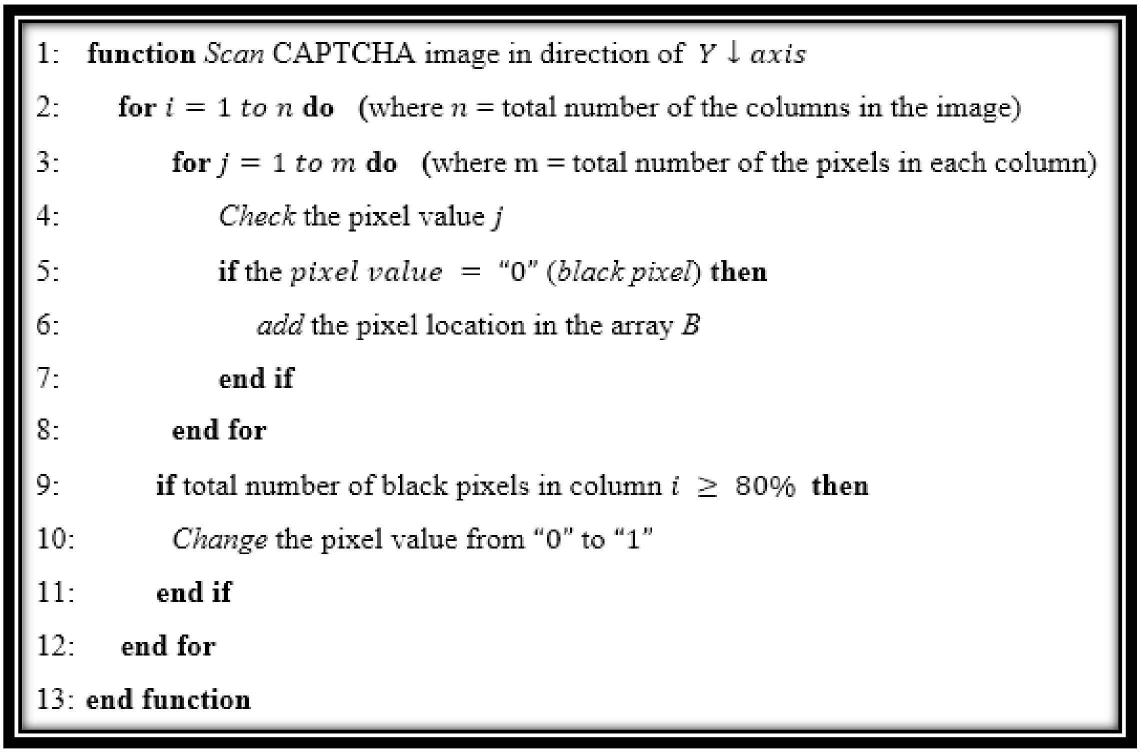


Figure 2.10: CAPTCHA image after line removal phase. As the image shows all the vertical and horizontal lines have been removed from the background [27].



```
function Scan CAPTCHA image in direction of $X \rightarrow axis$
    for $i = 1$ to $n$ do   (where $n$ = total number of the rows in the image)
        for $j = 1$ to $m$ do   (where m = total number of the pixels in each line)
            Check the pixel value $j$
            if the pixel value = "0" (black pixel) then
                add the pixel location in the array $A$
            end if
        end for
        if total number of black pixels in row $i \geq 80\%$ then
            Change the pixel value from "0" to "1"
        end if
    end for
end function
```

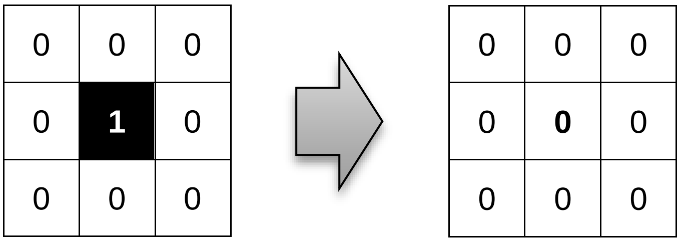
Listing 2.1: Horizontal line removal algorithm.



```
function Scan CAPTCHA image in direction of $Y \downarrow axis$
    for $i = 1$ to $n$ do   (where $n$ = total number of the columns in the image)
        for $j = 1$ to $m$ do   (where m = total number of the pixels in each column)
            Check the pixel value $j$
            if the pixel value $=$ "0" (black pixel) then
                add the pixel location in the array $B$
            end if
        end for
      if total number of black pixels in column $i \geq 80\%$ then
        Change the pixel value from "0" to "1"
      end if
    end for
end function
```

Listing 2.2: Vertical line removal algorithm.

- **Dot Removal Algorithm:** By applying line removal algorithms, there might be some black or white pixels in the background that act as random noise. To remove this noise, an algorithm is used that scans the entire image. If it finds any black or white pixel, it will then look at its eight neighbours as shown below.



0	0	0
0	1	0
0	0	0

0	0	0
0	**0**	0
0	0	0

In the case of finding a white pixel, the program will then look at all adjacent pixels. If all the neighbour pixels are white pixels, then it will turn that pixel to white and wise verse. The final output rendered result after applying dot removal algorithm is shown in Figure 2.11.

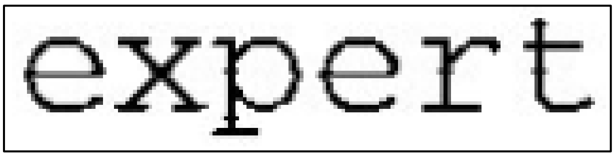


Figure 2.11: CAPTCHA image after the dot removal phase [27].

2. **Segmentation and Scaling:** In the Ez-Gimpy CAPTCHA model, all the characters are distant (or separated) so it is not a difficult job to segment the characters. The programme just follows the continuous pattern of pixels and where it stops is the end of the character. After segmenting all the characters, they are then scaled to the uniform size 60*40 pixel. The purpose of the uniform sizing of the character is to make recognition easier. As it has been shown in Figure 2.12, all the individual characters are separated and scaled to the standard size.

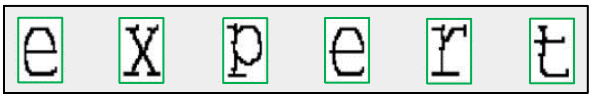
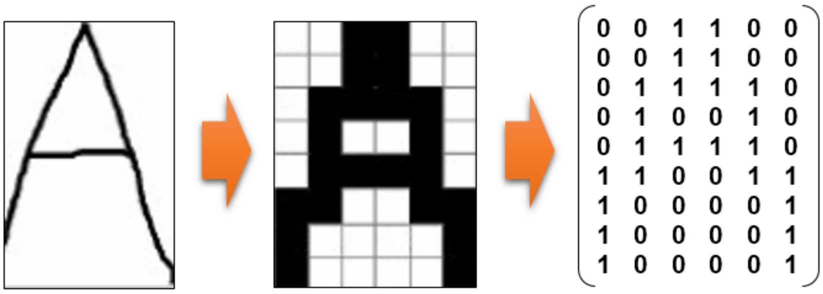


Figure 2.12: All the letters are being scaled after segmentation phase [27].

3. **Recognition and Pattern Matching:** The recognition phase involves comparing the extracted characters with stored characters on the server, and the character with the highest probability will be chosen as a candidate character. To get the model of the character, first, the I-matrix needs to convert all the black pixels to 1 and all the white pixels to 0 from the segmented characters, as shown in Figure 2.13.



$$\begin{pmatrix} 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

Figure 2.13: The process of converting a segmented character into a binary image, then converting it to I-matrix [27].

This process explains how a CAPTCHA image can be segmented and recognised using various recognition techniques. The final word '**expert**' is now easily recognisable after the computer programme has carried out this pattern matching process.

However, these algorithms can be easily modified to break most types of uniform background. As results of this research, it has been investigated how to remove the mesh background from the CAPTCHA image where the background is diagonal instead of straight lines as shown in the Figure 2.14. The solution to breaking this type of CAPTCHA would be to rotate the whole image by ***β*** degree, then the background mesh would become straight line as shown in the Figure 2.15. Then the vertical and horizontal line removal algorithm mentioned in the Listing-1 and Listing-2 will be used in order to remove the background lines from the image as it has been shown in the Figure 2.16.

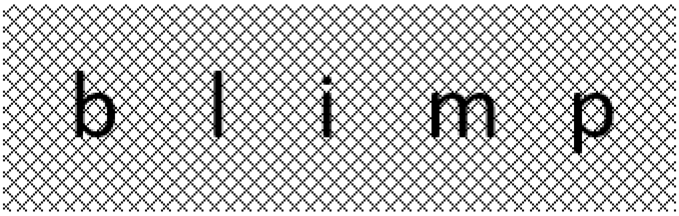


Figure 2.14: The Gimpy CAPTCHA image shown in the image has got mesh background with diagonal uniform shape instead of straight lines.

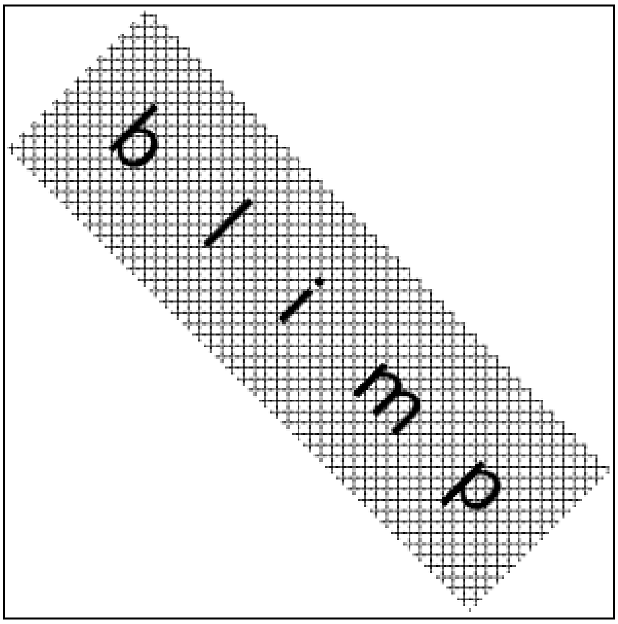


Figure 2.15: the CAPTCHA image is rotated by a ***β*** degree in order to make the mesh background straight vertical and horizontal lines.

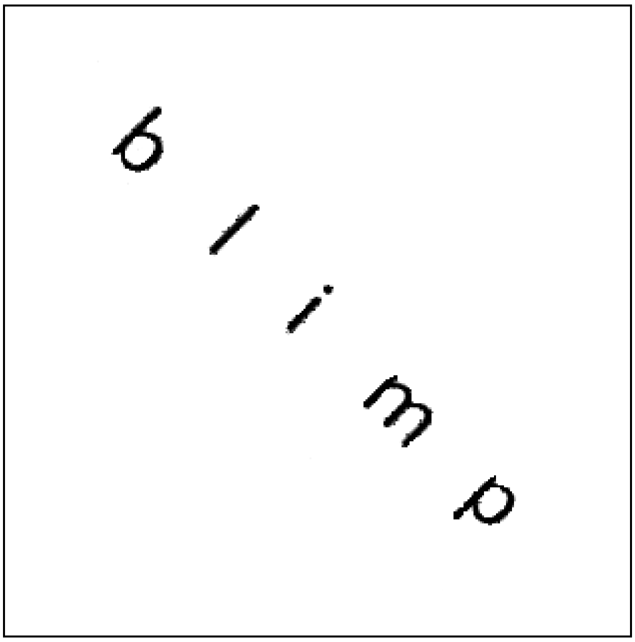


Figure 2.16: The final rendered output CAPTCHA image after applying "dot-removal" algorithm. As it is shown in the image, the background is removed using vertical and horizontal line removal algorithm.

After the pre-processing and removing the background noise from of the CAPTCHA image, then it will rotate by ***-β*** degree angle in order to bring the image to the first position. After the characters scaling and segmentation phase is complete, then the CAPTCHA will be processed by OCR system in order to recognise and decipher all the letters as shown in the Figure 2.17. As the image below shows, the word "blimp" is clearly distinguished and recognised using segmentation and character recognition techniques.

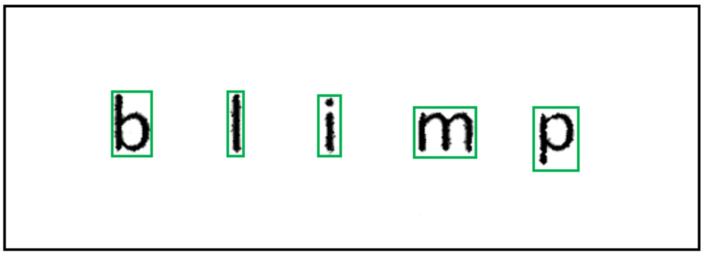


Figure 2.17: The CAPTCHA image is rotated by -β degree to bring it to the first position. As the image shows, the final characters can be easily recovered by applying segmentation and pattern matching algorithm.

As the results of this research project confirm, most of the uniform shape backgrounds can be detected and removed using diverse types of image processing algorithms as long as the prior information of the lines is known. Then the parameters of the lines can be estimated using, for instance, the *Hough Transform* for line detection.

Another case study would be to remove the colourful background from the text-based CAPTCHA using colour matching techniques since most of the current text-based CAPTCHA are printing with black ink for the text against a colourful background. An example of this kind of CAPTCHA is presented in the Figure 2.18. Therefore, by deploying colour matching algorithm it would be easy to filter or separate all the background colours apart from black colour. In this way, the background can easily distinguish from the foreground black text as shown in the Figure 2.19. After completing the pre-processing phase, scaling and segmentation will make the text readable for the OCR program as it is shown in Figure 2.20.

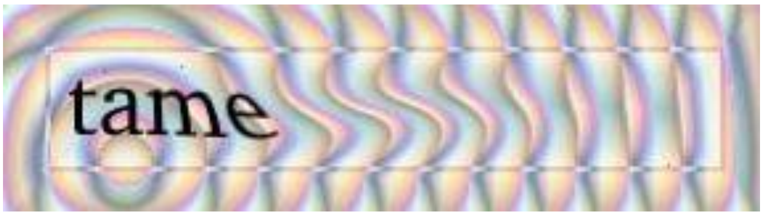


Figure 2.18: An Example of a CAPTCHA model in use at Yahoo! as it can be seen from the image, the text is printed in black against colourful background [28].

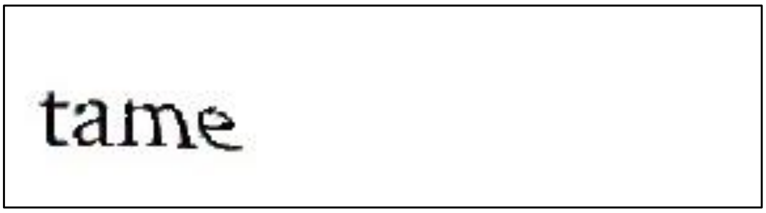


Figure 2.19: The final output rendered image. The background is completely removed using “colour matching” techniques and the black text is only remaining in the image.

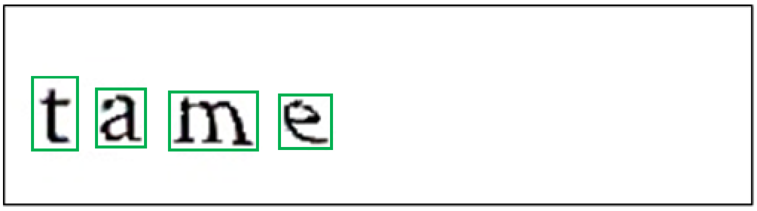


Figure 2.20: The figure shows after the background noise is removed, the text has been recognised using segmentation techniques.

From this evidence, we can conclude that dictionary-based information causes dictionary attacks on these types of CAPTCHA. Therefore, to have a more secure and robust model, one of the steps we must take is to avoid the use of a text string from a dictionary. In other words, if the presented information is random instead of dictionary-based, it will make it much more difficult for computer recognition programmes to decipher and recognise an entire CAPTCHA string [29, 30]. In the past, 3D-CAPTCHAs were assumed to be secure and robust because no OCR recognition software could break them directly. However, as shown in [22], 3D-CAPTCHAs have also been broken using different filters and image processing techniques with a high recognition success rate. Audio-based CAPTCHAs are also vulnerable to audio recognition techniques, as it is being discussed in [31].

2.7.2 Manual Third-Party Human Attacks

As discussed previously, the crucial point about CAPTCHAs is that they should be designed in such a way that makes them too hard for computer programmes to break, but should remain easy for humans to solve. CAPTCHA was initially introduced to distinguish between computer attackers and human users, but there is a critical question to be posed here: *what happens if the attackers are humans, not a computer?*

To break any CAPTCHAs, there are two possible methods. The first is to use sophisticated image recognition techniques that can recognise more complicated CAPTCHA images, yet this goal has become hard to achieve. As CAPTCHAs are getting more complicated than ever, using image recognition techniques to attack a CAPTCHA would involve complex multi-step image processing techniques, which in practice would be overly complicated and costly. However, the second solution, which is more reliable, is to use humans to break CAPTCHAs instead of computer robots. This kind of threat is dangerous [32]. To have a better understanding of how a third party human attack works, the process behind these types of attacks needs to be taken into consideration. As CAPTCHAs are designed to be easy for humans to solve, a guaranteed way to break a CAPTCHA is to use humans [33]. As research has shown, using cheap labour in different parts of the world is increasing every day. Cybergangs use cheap labour in third world countries at low cost to solve a vast number of CAPTCHAs every day [34]. As an example, in India, the market for solving CAPTCHAs is increasing very quickly, and workers can earn just $2 for every thousand CAPTCHAs they solve [35].

Computer attackers target different websites for different purposes. For example, they may buy thousands of tickets in seconds or alter the number of votes being cast in an online polling station. The computer programme will start by putting some invalid information into a server's online form. When it reaches the CAPTCHA stage to authenticate the user, the programme will send the CAPTCHA image to third-party human solvers across the globe, who will put the correct answer into the online form. The correct answer is then sent back via the same communication channel and the programme script will put the correct answer into the online form. This procedure is done through an application called IMCA (Instant Messenger CAPTCHA Attack) [33].

2.7.3 Comparison Analysis of Automated Attacks vs. Manual Attacks

De-CAPTCHA, which refers to the hacking of CAPTCHAs, is another important term in the CAPTCHA field of research, which will be discussed and explained later in this thesis. Regardless of the type of attack, it is, whether it be a manual attack such as a third party human attack, or an automated computer attack such as an OCR programme attack, there are different issues that need to be considered in order to ensure the optimal level of security. Table 2.1 presents a comparison of different attack types and their relevant characteristics, such as the cost of the attacks, the time it takes to design the challenge, and the time is taken to launch the challenge. The speed shows how fast the CAPTCHA challenge can be broken using either manual attacks or automatic attacks. As it can be observed from the table, using humans is the most efficient model to break any CAPTCHA challenge, as it has a much higher success rate than any OCR software. The results presented in Table 2.1 and Table 2.2 is based on the research I have conducted during this state-of-the-art on various CAPTCHA models in terms of efficiency and usability issues.

Table 2.1: A brief comparison of different criteria such as Cost, Set-up time, Speed, and Success rate for manual and automatic attacks.

Type of Attack	Cost	Set-up time	Speed	Success rate (Accuracy)
Manual Attacks (Third Party Human Attacks)	Very Low	Short (Seconds)	Slow	High (Over 90% Average)
Automatic Attacks (OCR-based De-CAPTCHA)	Very High	Long (Months)	Fast	Average (Around 40%)

2.8 Comparison Analysis of OCR and Non-OCR-Based CAPTCHAs

By analyzing and evaluating different CAPTCHA models including OCR and non-OCR-based CAPTCHAs, the results of this research show that different CAPTCHA models can be compared in terms of their conditions and characteristics. According to my research, one of the most crucial factors when designing any CAPTCHA type is time, whether this is the time it takes the user to solve the CAPTCHA challenge or the time it takes to create and design it. The popularity of different CAPTCHA models and how much brain power is used to solve the challenge are other critical issues when comparing CAPTCHAs. Table 2.2 shows a simple comparison between OCR-based, non-OCR-based, and non-visual-based (or audio-based) CAPTCHAs against different characteristics. Categorising the CAPTCHAs in this way makes it easier to find the most relevant and efficient model.

Table 2.2: Categorising the CAPTCHAs in terms of Difficulty, Security and Usability, Time to Solve, Setting-up Time, Popularity, and Use of The Human Brain.

Categorisation	Difficulty	Robustness (Security)	Usability	Time to solve	Set-up Time	Popularity	Using human's brain
OCR-based	Average-High	Low	Average-Low	Average-High	Very Short	Low	Average-High
Non-OCR-based	Average-Low	Average-High	Average-High	Average-Low	Average-High	High	Average-Low
Non-Visual-based	High	Average-High	Low	High	Low	Very Low	High

2.9 Comparison Analysis of the Current CAPTCHA Models

As it can be seen from Table 2.2, OCR-based CAPTCHAs are generally the easiest type of CAPTCHAs to make, but they are quite difficult to solve. On the other hand, generally speaking, non-OCR-based CAPTCHAs are quite difficult to build and they have a long set-up time, yet they are easier to solve and more user-friendly. They also take up less brain power, meaning that they require less effort and energy from human users to solve. As it can be observed from Table 2.2, non-visual (or audio-based) CAPTCHAs are the most difficult types to solve. They are more time consuming for a human user as the time to solve parameter is high. These factors cause audio-based CAPTCHAs to be unpopular with the public. By examining the above criteria, it is possible to formulate an alternative model that can overcome the weaknesses of current models. This new CAPTCHA model needs to be strong enough to resist against a range of attacks, yet also be user-friendly enough so that human users can solve it. Table 2.3 gives a brief overview of different CAPTCHA models as well as provides a short summary of what they do.

Table 2.3: Different classifications of CAPTCHAs with their relevant examples and brief explanations of each model.

CAPTCHA Classifications	CAPTCHA Name	CAPTCHA Type	Summary
OCR-based	GIMPY	Dictionary-based	Seven distorted words are selected randomly from a dictionary and at least three of them must be typed correctly by the user.
	EZ-GIMPY	Dictionary-based	One word is selected from a dictionary and an after distortion is presented to the user.
	ReCAPTCHA	pseudo-Random Word	One known word plus one unknown word is presented to the user.
	3D Interactive CAPTCHA	Random Character based	A cube is presented to the user and each face of the cube has got a different character. By rotating the cube, the user will see different characters.
	iCAPTCHA	Random Character based	A set of random characters is presented to the user and the user must click on the relevant character.

	3D CAPTCHA	Random Character based	Random characters are presented to the user in a 3D format with random background noise.
Non-OCR-based	**Bongo**	Object Recognition	A single image is presented to the user and the user must determine to which set it belongs.
	PIX	Object Recognition	A series of objects are presented and the user must choose the common name that describes all the objects.
	GWAP	Game	A series of games that the user must play in order to solve the CAPTCHA.
	Asirra	Object Recognition	A series of pictures of animals are presented and the user must choose only dogs or cats.
	GeoCAPTCHA	Scene Recognition	A scene of a natural place is chosen by the user and the user must remember the geographical information of that scene.
	Collage CAPTCHA	Matching	Some rotated objects are presented and the user must choose the right object.
	Jigsaw Puzzle CAPTCHA	Matching	An image is presented in the form of a puzzle and the user is required to swap two misplaced images.
	What's up CAPTCHA	Matching	The user must choose the correct orientation of an image.
	Avatar CAPTCHA	Face Recognition	The user must distinguish between real faces and avatar faces.
	Animation CAPTCHA	Video-based	A short movie clip is presented and the user is required to point out the key content of that movie clip.
	Face Recognition CAPTCHA	Quiz	An image of a very well-known person is presented and the user must recognize the face.
	Video CAPTCHA	Video	A short video clip is presented and the user must point out the key content or question in that movie clip.
Non-Visual-based	**Audio CAPTCHA**	Audio	Some characters and words with added background noise are presented to the user and the user must recognise these.

2.10 CAPTCHA Usability and Performance

Usability and performance are two key issues to consider when designing computer-based applications, including the design of CAPTCHAs. However, there is a trade-off between usability and security. When increasing the security of an application, the usability aspect will be adversely affected and vice versa. Therefore, in CAPTCHA research, whenever different security issues are discussed, the usability and satisfaction rate to pass the test are also considered. For these reasons, as part of this research, we will focus our investigation on CAPTCHA usability and performance by conducting a user study survey on one of the most popular CAPTCHA models, ReCAPTCHA [36]. This will be discussed in more depth in section 5.4. This CAPTCHA model is widely used on many websites, including Google and thus, understanding user perception of this model will help improve new CAPTCHA model designs. It can also facilitate the increase of the model's security level without reducing the level of its usability.

There have been several types of research in the fields of usability and performance of CAPTCHAs. However, they mostly concentrate on qualitative factors such as colour and distortion, as well as a limited number of quantitative factors such as accuracy and response time. When it comes to practical applications, a comprehensive set of tools is needed to ensure that usability and performance levels are measured, both quantitatively and qualitatively. According to Jakob Nielsen [37], the definition of usability can be defined by 5 qualitative elements: *Learnability*, *Efficiency*, *Memorability*, *Errors*, and *Satisfaction*.

The concept of learnability refers to how easy the test would be for a new user to cope with, and how easy the test would be for them to learn. Efficiency relates to how quickly a user could perform the challenge or in other words, how many attempts it would take for the user to give a correct answer (also known as accuracy). Memorability refers to how easy it would be for a user to remember the design of the CAPTCHA after completing it, and Errors refers to the number of mistakes a human user would make when attempting the challenge (also referred to as the response time). Finally, the concept of satisfaction refers to the level of enjoyment user experiences when participating in the challenge, or in other words, how willing the user is to participate in the challenge again. Accuracy, response time, and satisfaction are criteria that can be measured quantitatively and can, therefore, help us improve the usability of a new model. There are also other elements that can help us examine the usability of text-based CAPTCHA models as follows [38]: *Distortion* refers to the level of distortion on the characters and images, and their impact on CAPTCHA usability. *Content* is another very important element that examines the impact of the content of the CAPTCHA challenge on usability. These can include different criteria such as character set, string length, and random/dictionary-based words. *The presentation* is also another key issue to consider when examining CAPTCHA usability. This includes factors that can affect the presentation of the challenge to the user such as font type, font size, image size, use of colour, and the way the CAPTCHA challenge is integrated into the webpage. Table 2.4 presents different key elements that can have an impact on the usability of CAPTCHAs.

Table 2.4: CAPTCHA usability issues.

Elements	Usability issues
Distortion	Character Distortion
	Vague Characters
	Language
Content	Character Length
	Random Character -based
	Dictionary-based
Presentation	Font Type and Size
	CAPTCHA Image Size
	Use of Colours
	Website Integration

A. Distortion Issues: Character distortion, as explained in [38], can significantly affect CAPTCHA usability because the level of distortion can directly affect readability or the capability of a user to recognise and read the characters. It is indeed the case with some applications, such as ReCAPTCHA, which is used by many major websites, that the level of distortion on the text is so high that users may be unable to recognise them from the image, and multiple attempts may be necessary to solve the challenge. An example of such levels of distortion is shown in Figure 2.21.

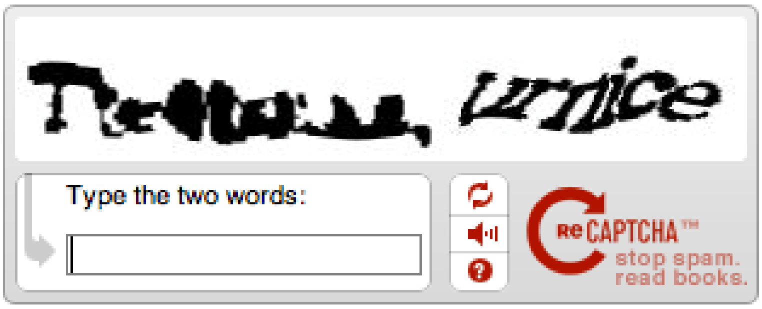


Figure 2.21: An example of an unreadable CAPTCHA due to the extreme level of distortion [39].

The presence of vague letters and characters is another type of distortion, which will cause ambiguity in recognising the displayed characters in a CAPTCHA image. Some characters have a distinct shape that the distortion may change considerably, so much so that the character looks completely unrecognisable. Figure 2.22 presents some examples of over ambiguous CAPTCHA images [38].

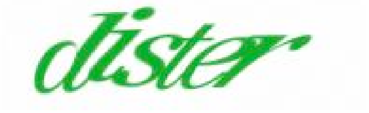
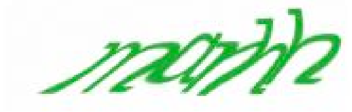
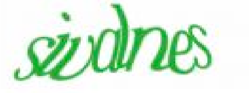
CAPTCHA	Vague Characters
	The first character could be ‘d’ or ‘cl’
	The first character could be ‘m’ or ‘rn’
	The second character could be ‘w’ or ‘iu’

Figure 2.22: Examples of over ambiguous CAPTCHA characters.

Another relevant example of over distortion in a CAPTCHA image can be seen on the MSN website. In this CAPTCHA model, random arcs obscure the distorted characters, which may cause ambiguity for users, as shown in Figure 2.23. As it can be observed, it is unclear whether the first character is letter ‘j’ or just a random arc.

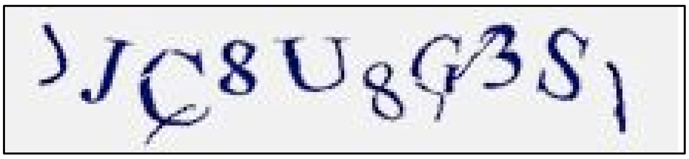


Figure 2.23: MSN CAPTCHA where it is not clear whether the first letter is a ‘j’ or just a random arc.

Choice of language in CAPTCHA images is another crucial factor to be considered. CAPTCHAs are used worldwide and therefore their users are all from different nationalities and speak different languages. There has been researching in this field, such as [38], which shows that, for users whose first language is not English, recognition success rates are much worse than native speakers. Moreover, according to the published statistical results from the recent ReCAPTCHA model by von Ahn, recognition success rates for English speaking countries is about 97%, yet recognition success rates for non-English speaking countries is about 93%. This study clearly demonstrates that CAPTCHA recognition success rates directly depend on the language that people speak in their part of the world.

B. Content Issues: The length of the character strings that make up a CAPTCHA image can also have a direct effect on both its security and usability. Indeed, the longer the string length, the lower the risk of random guessing attacks. On the other hand, longer character strings also reduce the usability of the test because it becomes harder for a user to recognise the characters. To explain this more clearly, if one assumes that the recognition success rate for every individual character is r where $r < 1$, the recognition success rate for an entire string made of n characters is r^n. Thus, the recognition success rate decreases with the value of n [38].

The choice as to whether a CAPTCHA image is made up of random characters or a single meaningful word has a direct effect on the usability and security of the challenge. If random characters are used to make up a CAPTCHA challenge and the longer the length of the string, the harder it would be for a user to decipher them. However, according to the research by ReCAPTCHA, the longer the length of the CAPTCHA, the easier it gets for people to recognise a single word [6]. According to *Gestalt Psychology* [40], it is much easier for humans to recognise an object as a whole, rather than recognise its parts. Saying this, choosing a meaningful word for the CAPTCHA test would make the image easier for humans to understand and decode Therefore, in practice, this will also increase the risk of computer recognition attacks. There are numerous ways of decoding dictionary-based CAPTCHAs, such as dictionary attacks and pixel count attacks [41], [42].

C. Presentation Issues: Another crucial factor that can affect the usability of CAPTCHAs is the way that the challenge presents its contents to the user. More specifically, the type and size of the font, which is used to generate the CAPTCHA, can cause readability and recognition issues for the user [43], [44]. The choice of size of the CAPTCHA characters could affect usability, as was suggested in recent research [14], which carried out a comprehensive analysis of character size in several CAPTCHA models. Another key factor that can affect a CAPTCHA's usability is the use of colours. Using a colourful interface decreases the usability and may potentially harm the security of the test [45]. This is due to the physiology of the human eye as the retina contains two types of photoreceptors: rods, which are sensitive to stimuli of different shades of grey (including black and white), and cones, which are colour sensitive [46]. In healthy human eyes, there are around 120 million rods and about 6 to 7 million cones, with rods being a thousand times more sensitive than cones [47]. Due to this, human eyes are more sensitive to black and white, rather than colour images. Indeed, this is one of the reasons that our new proposed CAPTCHA model uses black and white colours.

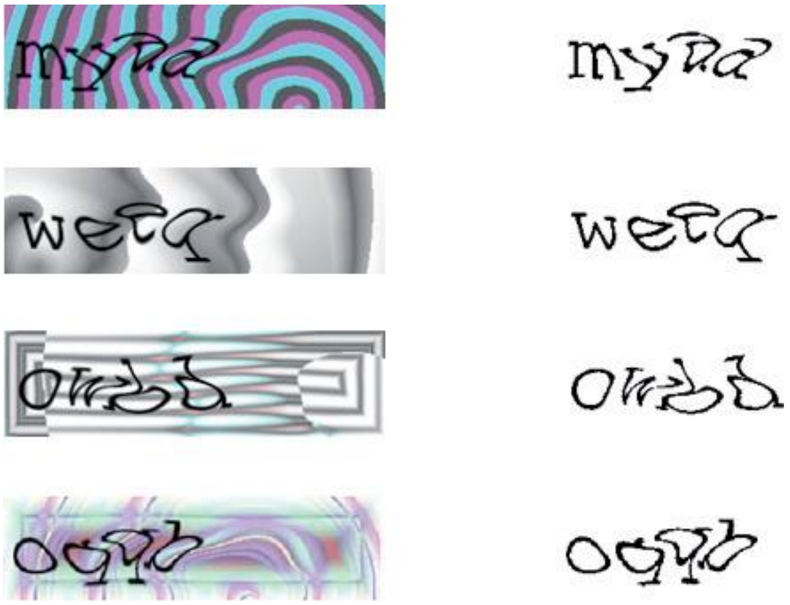


Figure 2.24: Using the colour matching algorithm to break the Gimpy CAPTCHA challenge. As the examples showing, using distinct colour for the text and background noise would make it easy for the OCR to filter the colours.

As stated, not only can using distinct colours in the CAPTCHA image decrease usability, but it can also be risky in terms of security because most text-based CAPTCHAs are broken using segmentation and recognition techniques. Using varied colours for the background and foreground could increase the risk of an attack based on Colour Filling Segmentation (CFS) [42], which separates different colour information in order to extract the foreground from the background. As shown in Figure 2.24, the colour matching algorithm is used to extract the black and white foreground from the colourful background [45]. Another example of this is BotBlock, which uses a sophisticated colour management algorithm for the background and foreground. As can be seen in Figure 2.25, the CAPTCHA is made up of random shape blocks with random colours in the background, which also appear in the foreground [45]. This technique can pose serious challenges for users, particularly in the case of visually impaired people. These types of CAPTCHAs can also be broken using sophisticated colour matching techniques and the pixel count attack method, as mentioned in [45].

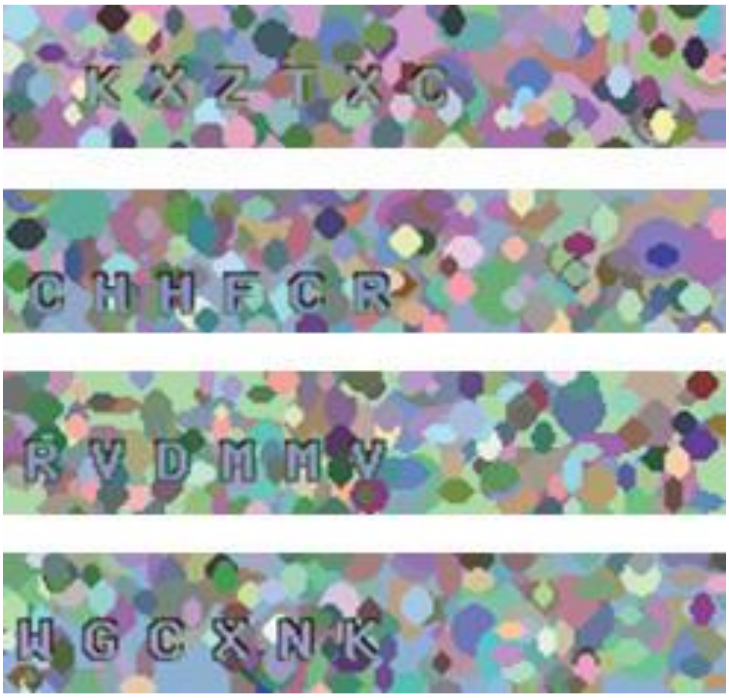


Figure 2.25: Example of the BotBlock CAPTCHA. Using the same colour for the test and background noise would make the CAPTCHA difficult to recognise for the human users.

Finally, another factor to consider is the way in which the CAPTCHA challenge is integrated into a webpage. The location and position of the challenge are very important because it must be easily visible and accessible to users. In some cases, there were issues with the typing-in box, such as the ReCAPTCHA model, which required users to activate the input box in advance [38]. These requirements may cause frustration and dissatisfaction, which may decrease the usability of the CAPTCHA.

2.11 Conclusions

To conclude, the main objectives of this chapter were to present a comprehensive analysis of CAPTCHA models, to show how important the security of online service providers is, and to also outline the possible threats that can put our personal information at risk. Automated computer programmes are stealing information and abusing online systems by imitating human users. As discussed, CAPTCHAs are being widely used to distinguish between real human users and computer bots. As the results of this research show, OCR-based CAPTCHAs are most vulnerable to a range of attacks, and the vast majority of OCR-based CAPTCHAs have been broken using sophisticated OCR software. As results of that, IT and cybersecurity experts announced that newly developed CAPTCHA breaker software called “DeCaptcha” can break audio CAPTCHAs up to 89% success rate. Also as the results of researches showing, current Text-based CAPTCHA models have been broken with high success rate. For instance, eBay CAPTCHA was broken at 82% and Microsoft CAPTCHA was broken with 42% success rate [8]. On the other hand, non-OCR-based CAPTCHAs are very time consuming as well as being very costly to create and build. Additionally, the most important types of CAPTCHA attacks associated with each CAPTCHA type have also been studied and analysed.

All of the CAPTCHA models mentioned in this chapter are based on the perceptual abilities of humans, both visual and auditory. However, many people who use the internet today may have impaired hearing or vision, which means these methods are not effective enough to cater for all types of people. For example, some people have problems recognising different colours, and others cannot hear noise levels properly. Another critical issue CAPTCHAs face is their language dependency, which may not be accessible in many parts of the world. Finally, there is also an issue regarding the quality of the CAPTCHA characters or sound clips because many are often over ambiguous and virtually impossible to decipher, dramatically decreasing the usability of the CAPTCHA challenge. The next chapter will focus on one of the most advanced abilities of human users: namely, the ability to superimpose and integrate perceived visual information using the process of IM, an ability that is totally unique to humans. Thus, this thesis suggests that none of the current computer recognition techniques would be able to understand or decipher a CAPTCHA that utilises this technique, making our new proposed model very robust.

Chapter 3

A Visual Psychophysics Approach to the Proposed CAPTCHA Model

3.1 Introduction

The previous chapters have defined what is meant by a CAPTCHA, discussed the importance of online security, outlined diverse types of CAPTCHAs in regard to their specifications, and also presented the main security aspects of different CAPTCHA challenges. As has been emphasised, a CAPTCHA challenge should be designed in a way that makes it easy for human users to solve, but difficult for computer recognition programmes to break [48]. In this chapter, a novel approach has been introduced called persistence of vision which is based on the unique ability of human eyes to superimpose and integrate all the seen frames using trans-saccadic integration techniques. Since the characteristics of IM (Iconic Memory) is known only available to human eyes, therefore it is believed that a CAPTCHA model based on this phenomenon would be also robust against most current computer recognition techniques.

3.2 Persistence of Vision and CAPTCHA

Currently, neuroscientists and psychologists believe that the process that enables us to see the world as integrated and continuous is a phenomenon called POV. This is the core reason why the world around us does not turn black with each blink of our eyes [49]. POV is a key component in any movie produced by the film industry as it enables humans to see a film as a smooth-running series of moving images. Every film is made up of a sequence of individual fleeting images (or frames). By running these in front of the human eye, POV will cause the illusion of integrated and uniform shapes into our visual system. Since POV is a unique characteristic of the human eye, we have utilised this distinctive ability to distinguish between real human users and automated computer bots in our new CAPTCHA model. To gain a better understanding of how POV works, we first need to consider the main causes of this phenomenon. What is known as ‘afterimage’ causes our visual system to remember the effect of every single image we see for a very brief period in our IM, following the disappearance of the object from our vision [50]. This persistence can last for one-tenth to one-fifteenth of a second depending on different criteria such as image brightness, colour, and the angle of light [51].

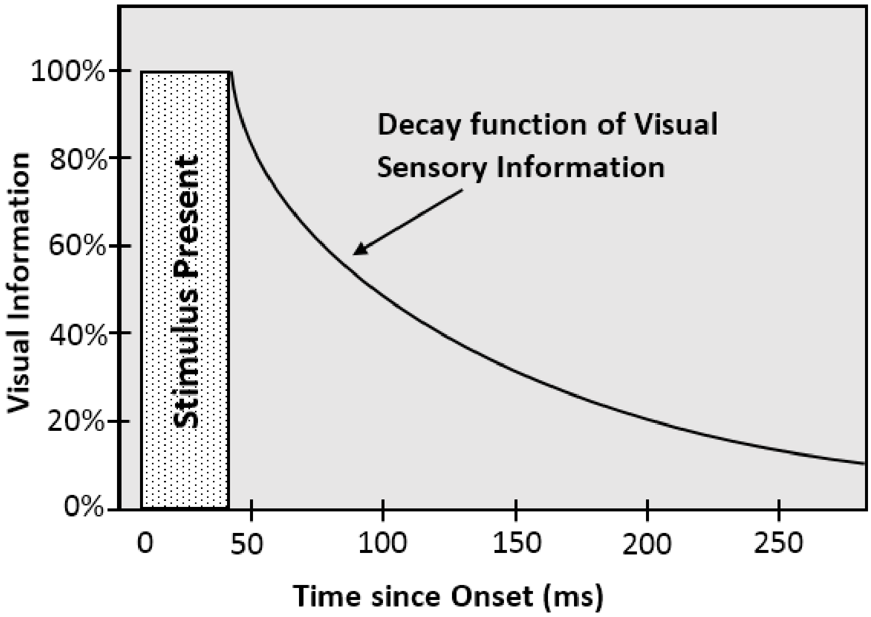


Figure 3.1: The graph represents the rapidly decaying function of visual sensory information following the stimulus offset [52]. The sensory information will remain until about 100 to 150 after the stimulus offset and then starts to wipe off from our memory.

According to research, the afterimage is the cause of POV in the brain. Studies dictate that healthy human eyes cannot react or distinguish changes in light frequency in the visual system any faster than a certain period. Thus, the final outcomes will either not be noticeable to the human eye, or the changes in light frequency will be seen in an integrated form [53]. As shown in Figure 3.1 [52], when a stimulus is present, the human visual system can pick up the most information. However, this visual sensory information will drop gradually after the stimulus disappears from our vision. As it can be seen in Figure

3.1, the quality of visual sensory information is at its maximum level within 0-50 milliseconds after the stimulus offset and then it decreases rapidly. This quickly decaying function can explain the fundamentals of IM and POV, as elaborated in [53].

3.3 Persistence of Vision and The Film Industry

As stated earlier, POV is a significant component that enables humans to watch films. Every movie is made up of lots of single images (or frames) that pass before our eyes at high speed. This creates the illusion that the objects and shapes are moving. There has been a lot of research in this field, and studies have concluded that the total number of images (or frames) that pass in front of our eyes can make an enormous difference to the quality of the video. The higher the number of images passing per second, the smoother the video will be. My research shows that any frame rate that is less than 16 frames per second (FPS) will cause the human eye to see flashing images, instead of a smooth-running video. However, in some publications is has been mentioned that even at a frame rate of fewer than 10 FPS, the motion will still be understandable for the audience. In this case, the film itself can be seen and understood, but will just not be as smooth as a movie played at a higher speed. Flipbooks are a good example to explain this phenomenon.

It can be concluded that the more frames produced per second, the better quality the movie will be, but at the same time, the total size of the movie will also increase. Higher resolution movies have got bigger in size and thus, from a consumer point of view, they will need more space to store. Modern movie technologies run at 24 FPS, which is the standard rate for movie theatre film projectors. A total of 48 frames per second will produce slow-motion movies [54]. Faster rates, such as 300+ FPS, are also used in high-speed cameras during sporting events, which require a rapid change of scene [55]. As it can be observed from Figure 3.2, when black and white images are placed consecutively together at high speed, POV will cause the illusion in our vision system that the horse is running. These images were taken by Gordon McConnell [56].

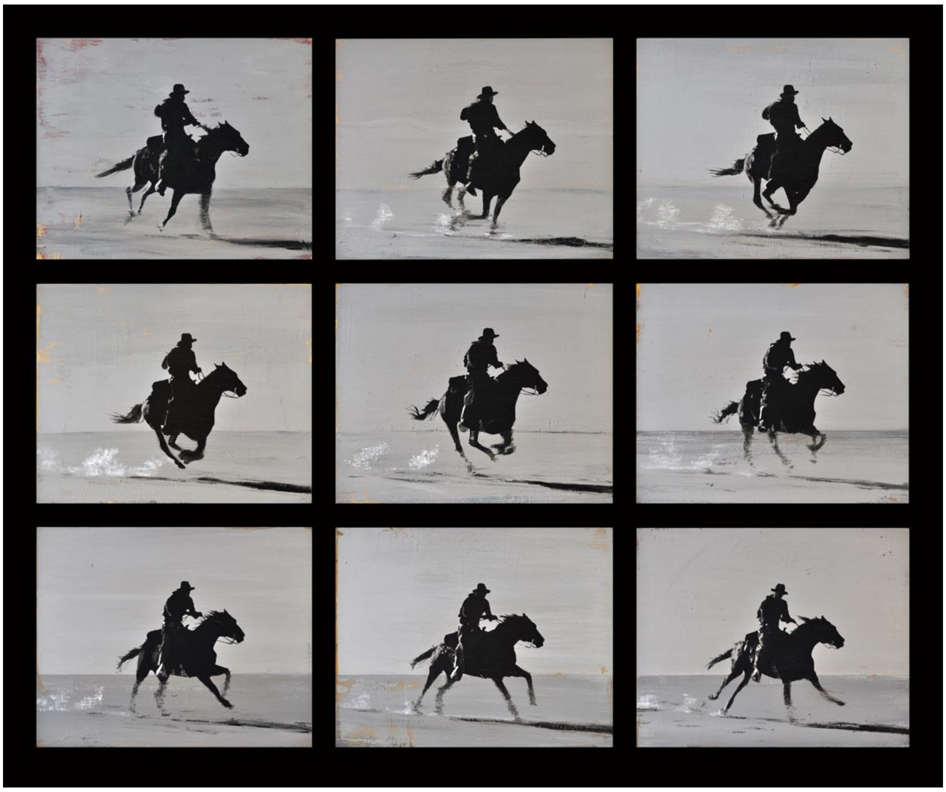


Figure 3.2: An example of POV: a series of images playing consecutively to create the illusion of movement.

3.4 The Effect of Persistence of Vision in Temporal Integration

Following this brief introduction to the procedure of POV and the functionality of IM, this thesis will now explain how POV can cause the HVS to see the world in an integrated and uniform fashion. To explain this, first, I need to refer to the definition of POV. As the light from a stimulus hits the retinal part of our eyes, the effect of the light can be retained in our visual system for a brief fraction of a second before disappearing. If a second stimulus is presented whilst the visual information of the first stimulus is still retained, the visual system will perceive the two stimuli as a single stimulus. In psychophysics, this phenomenon is known as Temporal Integration (TI). According to [52], the visual information for each stimulus can persist for about 100-200 milliseconds after its offset. However, this persistency can be affected by a number of factors, such as stimulus intensity, duration, and colour. TI is known as the effect of two stimuli appearing with a very short delay, or ISI from each other. In other words, ISI is defined as the distance between the offset of the first stimulus and the onset of the second stimulus. If the two stimuli are presented with a long ISI delay, then the visual sensory information for the first stimulus will already be wiped from our vision. Therefore, there will be no integration happening with the second stimulus, as shown in Figure 3.3. However, when presenting the two stimuli with a very short ISI delay, a person will be able to see the results of the two signals as one integrated signal, as shown in Figure 3.4.

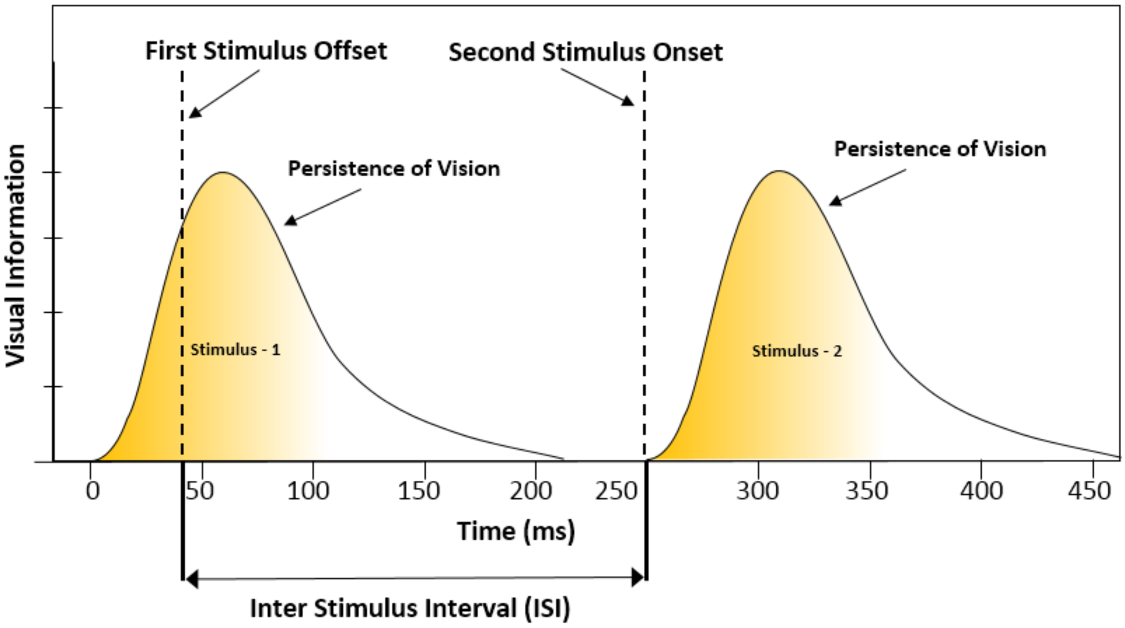


Figure 3.3: Two stimuli are presented with a long ISI delay. As shown in the graph, there is no overlap between the two signals. This means that visual sensory information from the first stimulus is wiped completely before the presentation of the second stimulus.

As discussed previously, after the stimulus offset, some of the sensory information will remain in the HVS for a very brief fraction of a second before completely disappearing (known as POV). However, if the second stimulus is presented whilst the effects of the first stimulus remain, the HVS will superimpose the two signals together and perceive them as one integrated image. The new image will contain characteristics of both stimuli, known as TI [52, 53]. The properties of this phenomenon are being used for the development of our proposed CAPTCHA model.

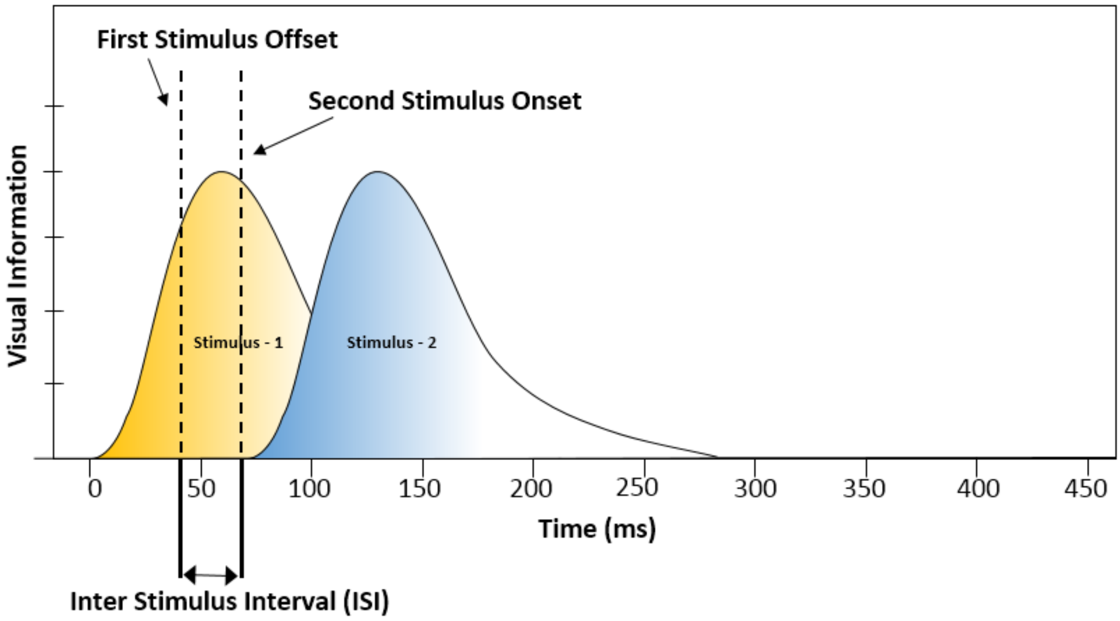


Figure 3.4: Two stimuli are presented with a very short ISI delay, and thus, there will be some areas where the two signals overlap each other. The overlapping areas will have some information from stimulus-1 and some from stimulus-2.

3.5 Trans-Saccadic Visual Integration Technique

As has already been acknowledged, POV causes TI, which subsequently allows us to perceive the surrounding world in an integrated and continuous fashion. To appreciate how the mechanism of TI works, a novel scheme for it is proposed in this section, which will help us to understand how a sequence of images (or frames) are combined by the brain using trans-saccadic eye movements. According to research, healthy human eyes are characterized by rapid eye movements, occurring about 3 to 5 times per second, which are known as saccadic eye movements. These last on average for approximately 30 milliseconds [57, 58]. These rapid eye movements are necessary for our visual system to perceive a high-quality image from the surrounding environment through integration and fusion of visual information. During trans-saccadic eye movements, there are intervals called *fixations*, which last for approximately 300 milliseconds [58]. During each fixation period, the foveal part of the eye focuses on an object and sends the visual information of the object to the brain to analyze and process. Fixation periods are separated by rapid eye movements, called *saccades*. During each saccade, the visual information perceived during the fixation $f^{(i)}$ is combined and superimposed with the visual information from the previous fixation $f^{(i-1)}$. This integration procedure takes place in a part of the memory module known as the trans-saccadic memory [58] [51]. Yet, in many studies, the trans-saccadic memory is said to have the same characteristics as the VSTM [53, 59]. According to [58], human working memory can process information of 3 to 4 saccades at one time. Therefore, to build a stable visual impression of the environment (or a scene), repetition of the visual information is required. To better explain how the proposed trans-saccadic integration scheme works, the following section will consider this scheme under two scenarios.

3.5.1 Single Stage Scenario (SSS)

As discussed previously, to watch a video clip smoothly without flashing images, all the images need to be presented in such a way that it enables our visual system to integrate and superimpose them. To achieve this goal, the visual information perceived during one fixation period needs to be combined with the visual information perceived during the subsequent fixation period. Figure 3.5 [49] shows the proposed processing scheme for trans-saccadic integration, which takes place in a human's short-term memory. The proposed model starts by perceiving visual information from our environment during a fixation period $f^{(i)}$. This fixation period lasts for about 300 milliseconds and is known as *visual information acquisition*. All the visual information received during each fixation period is stored into the IM for a very brief period before the information passes into the short-term (or working) memory. With each saccade, the visual information stored in the IM will be passed to the working memory to be integrated with the pre-stored visual information from the previous fixations, $\lambda g^{(i-1)}$. In visual psychophysics, this procedure is called *trans-saccadic integration*. Another important procedure that takes place during each saccade is the erasing of the IM to prepare it for receiving new visual information from the eye.

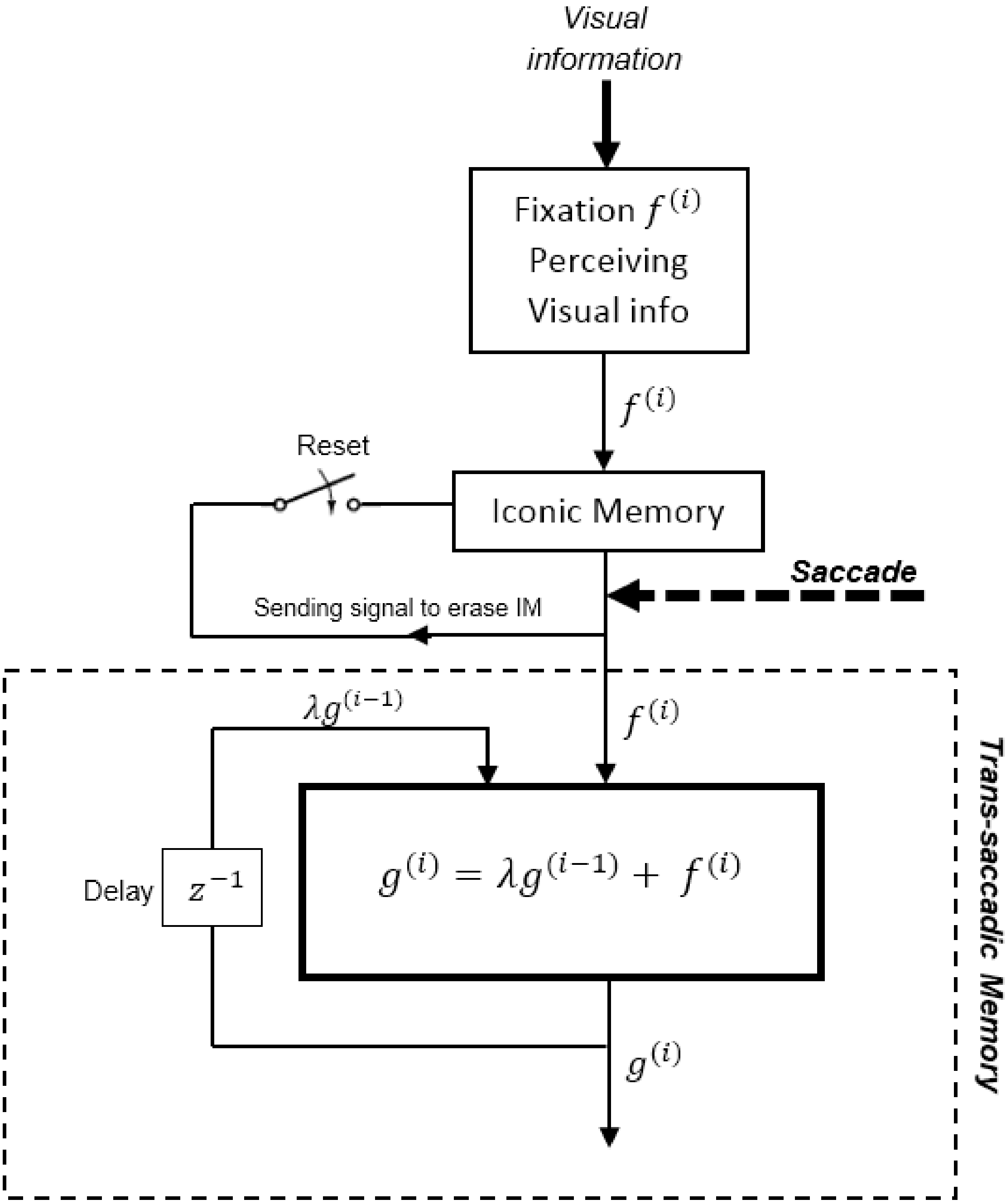


Figure 3.5: The procedure of visual information integration using Trans-saccadic memory. Visual information will perceive during each fixation in our iconic memory. Each saccade will cause our sophisticated visual system to superimpose all the previous perceived visual information with the new visual information [49].

3.5.2 Multi-Stage Scenario (MSS)

As can be seen in Figure 3.5, to see the world in an integrated and continuous fashion, the process of TI needs to be repeated. The output image $g^{(i)}$ is the result of superimposing a series of frames, which were perceived during the previous fixation periods, multiplied by a weight, known as *forgetting factor* λ, which can be described as follows:

$$g^{(i)} = \lambda g^{(i-1)} + f^{(i)} \tag{3.1}$$

$0 < \lambda \leq 1$ is the forgetting factor that will allocate exponentially less weight to the older samples. The term $\lambda g^{(i-1)}$ refers to the weighted outcome of the previous integration of the fixations. In the model, older samples are allocated a lower weight and therefore, the influence of their visual information will automatically diminish. By iteratively expanding Equation (1) and substituting the associated terms, we obtain the formula below for several n fixations:

$$g^{(n)} = \sum_{i=0}^{n} \lambda^{(n-i)} f^{(i)} \tag{3.2}$$

Equation (2) shows how fleeting images will produce the final image in the visual system using the trans-saccadic visual integration technique. As we can see, a consecutive frame sequence of A = $\{F_1, F_2, F_3, \ldots, F_n\}$ is being run at high speed and the HVS would be able to integrate and superimpose all these fleeting frames during different fixation periods in order to produce the final image. Since every single image (or frame) will be retained in our IM for a very brief period before being wiped, and should be repeated many times for our memory system to remember and memorise the sequence for a very brief period (in milliseconds). This process of repeating images causes our visual system to distinguish between variations of pixel frequencies. Consequently, the user will recognise the final image combination.

Every single frame is made up of several random pixels, some of them belonging to the object, while others are background noise. As mentioned earlier, this approach uses several consecutive frames (F_1, F_2, F_3 ...,) in the sequence 'A.' These are run rapidly, and every single image will be retained in the visual memory before disappearing, due to POV. This will cause the human visual system to combine all images and then 'reconstruct' the final image, which is the superposition of the previous sequence.

3.6 Conclusions

POV can be defined as a concept of psychophysics, which refers to the unique ability of the human eye to remember every single image after the main source of light has disappeared from sight. This phenomenon was the main inspiration to develop a new CAPTCHA model that will work only for humans, and not computer programmes, in order to tell the difference between them (as is the purpose of CAPTCHAs). This robust CAPTCHA model relies on the unique human ability of perception, something that no current computer programmes would hope to have. Therefore, it is possible to create a CAPTCHA model that no current character recognition programme would be able to break, as the test would only be meaningful to humans.

In this chapter, the emphasis has been placed on psychophysics and more specifically, the ability of the HVS to produce an image based on a collection of the fleeting image using the IM. In addition, it has been concluded that the HVS is a sophisticated entity, a system that computer programmes cannot compete with. Furthermore, in this chapter, our new mathematical CAPTCHA model, which uses a trans-saccadic visual integration technique was introduced. This is a novel CAPTCHA model based on POV, one that is only solvable for human users and not bots. This integration model would work based on single and multi-stage scenarios.

The next chapter will focus on this new CAPTCHA model, which is entitled VICAP. This model utilizes the process of superimposing and integrating visual information using trans-saccadic memory. The advantages and disadvantages of this new technique will be also discussed in the following chapter.

Chapter 4

Proposed Visual Integration CAPTCHA Model (VICAP v.1)

4.1 Introduction

In this chapter, a novel CAPTCHA model is introduced labelled VICAP, which is based on psychophysics and the properties of the human Visual Short-Term Memory (VSTM). This method would superimpose and integrate a range of fleeting frames of visual information, which would then be captured by the human eye and pieced together into a final image in the user's brain. This proposed model is designed to capitalize on a user's sophisticated visual abilities. Therefore, it would be logical to conclude that this proposed CAPTCHA model would be robust against computer recognition programmes, which do not possess this uniquely human ability.

4.2 Introducing Visual Integration CAPTCHA (VICAP) Version.1

As discussed, the proposed CAPTCHA model is based on the sophisticated ability of the HVS to retain visual information about frames passing quickly before the human eye, even after a light source has disappeared for a very brief period (known as IM). Hence, the brain would be able to combine all the perceived visual information in order to create a final image. This ability is unique to the human perceptual system and therefore, it is meaningless to current computer recognition software.

Using uniform and regular patterns in the background and foreground objects will make the pattern easy to recognise by using basic pattern recognition techniques. In short, it would be easy to remove the background from the object and rebuild the distorted characters. Additionally, using a variety of colours for the object and background would make it easy to distinguish between the background noise and the object pixels. For this reason, in this new proposed CAPTCHA model, background pixels and object pixels have been randomised and sampled to make it almost impossible for computer recognition programmes to guess and reconstruct patterns. The sophisticated HVS and brain would easily be able to build the image of the characters. The CAPTCHA image will be fleeted at high speed in front of a human user's eyes in order to create a picture of the final object.

4.3 CAPTCHA-Generator Application Overview

To generate the VICAP output frames sequence, the CAPTCHA-Generator Application (CGA) was developed using .NET programming tools, running on a 64-bit Windows operating system with Intel Core-i5 CPU and 3.20 GHz processing power. The key role of the CGA is to render individual VICAP images based on the specific criteria, which will be discussed below, and play them back in a sequence of frames consecutively and smoothly for users, producing a film (or animation) effect.

The VICAP model uses a combination of five characters and numbers. Since this combination is selected on a random basis, it will not be possible for computer programmes to guess, as is the case with dictionary-based attacks. The string of characters to be used by the CGA is made up of 18 letters as follows: {A, C, E, F, H, K, L, M, N, P, R, S, T, U, V, W, X, Y} and 6 numbers as follows: {3, 4, 5, 6, 7, 9}. In total, there will be 24 different characters and numbers to be randomly selected by the programme. To make the characters easier to see, some letters and numbers have been avoided due to the ambiguity they may cause. For example, in many cases, the letter 'B' can easily be mistaken with number '8' and vice versa. Another case is the letter 'O', which can cause ambiguity with the number '0'. The letter 'Z' can also be confused with the number '2' and so on. For more information on CAPTCHA usability issues, please refer to the work by [60]. To generate every single VICAP frame and display it to the user as a video clip, the following six steps need to be taken, as presented in Figure 4.1.

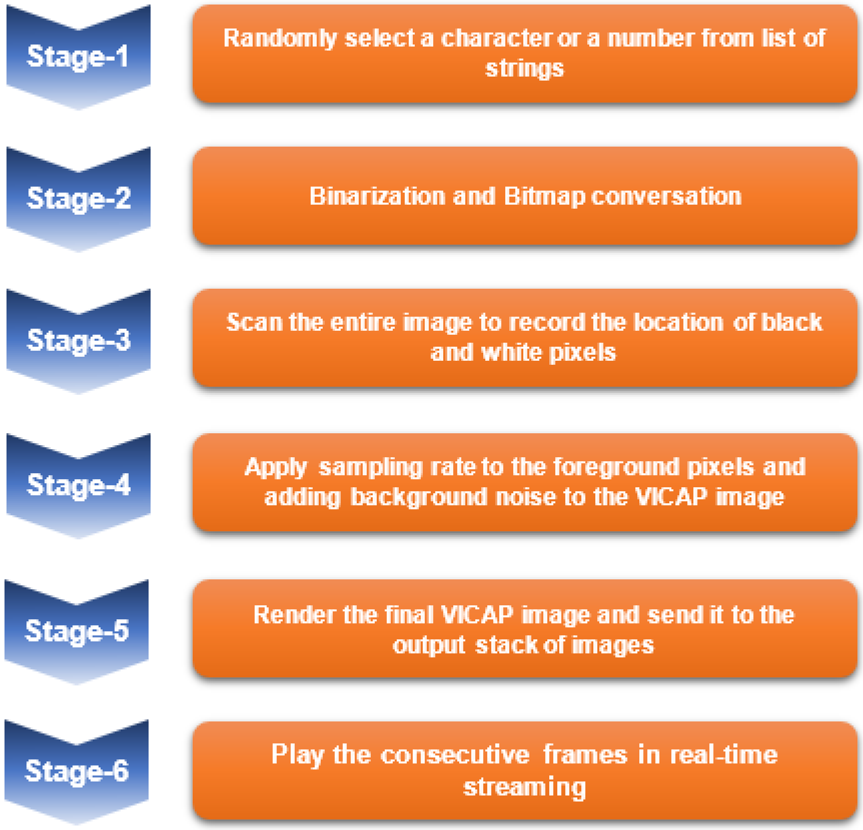

Stage-1
Randomly select a character or a number from list of strings
Stage-2
Binarization and Bitmap conversation
Stage-3
Scan the entire image to record the location of black and white pixels
Stage-4
Apply sampling rate to the foreground pixels and adding background noise to the VICAP image
Stage-5
Render the final VICAP image and send it to the output stack of images
Stage-6
Play the consecutive frames in real-time streaming

Figure 4.1: Proposed CAPTCHA model development plan (version 1). Different steps need to be taken in order to generate and play consecutive VICAP frames for the user.

One of the most common problems with text-based CAPTCHAs is that the user often cannot read the text properly. There is no option for the user other than refreshing the challenge to gain a new CAPTCHA test, which often brings the same problem because the challenges are generated by random fonts. One of the key features that make our proposed model different from other CAPTCHA types is that the user can select what type of font he or she likes more, and the CAPTCHA challenge will use this font of choice. This makes the test more user-friendly because they will be reading a font that is familiar to them. Since each user has a diverse set of fonts on his or her computer, it was not possible to predetermine any specific fonts because they may not be available on the user's computer and the CAPTCHA would not work. For that reason, a function has been created that downloads all available fonts from a user's computer and presents them to the CAPTCHA application. This makes the CAPTCHA very flexible as fonts vary dramatically from computer to computer. In modern digital typography, the term 'Font' is also known as 'Typeface.' Each font or typeface is made up of different individual symbols and characters known as ‘Glyphs.’ There are two main types of digital computer fonts, as follows:

1- **Bitmap Fonts:** These types of digital fonts are made up of small grades of dots and pixels, which represent each glyph or character. Bitmap fonts are not scalable and therefore, there is a need to have different images associated with different typefaces, characters, and font sizes. For these reasons, they are not very efficient in terms of storage because they take up a lot of memory.

2- **Outline Fonts:** These types of digital fonts are made up of a combination of lines and curves that represent each glyph or character. The key point about outline fonts is that they are scalable, which means they can be scaled to a variety of sizes without pixelating. Since these types of font are made from complicated mathematical algorithms, they are more sophisticated in rendering and processing.

Almost all computer fonts that are being used today are made of ‘TrueType’ fonts. TrueType fonts are usually compared with Bitmap fonts, which are scalable and can be printed in any size, scale, or format. Even though Bitmap fonts have only been created in certain sizes, for each character size and each font type there will be individual sets of images representing individual characters and glyphs. In the late 1980s, computer graphics company Adobe introduced a specific font type called ‘Type-One’ fonts, which are based on ‘Vector Graphics.’ Vector graphics are made up of different mathematical expressions to represent lines, curves, and polygons. That means it would be able to resize and rescale the font without losing any quality in the appearance. Later, two companies, Apple and Microsoft, created a very similar method called ‘TrueType’ technology, which can rescale the font and make it bigger or smaller without reducing its quality. TrueType technology itself is made up of two different elements: 1- The True Type Rasterizer and 2- True Type fonts. The Rasterizer is a piece of software that translates the mathematical data that has been used to create each font, including characteristics such as size, colour, orientation, and location of the font. This data can be readable for computer graphics cards and monitors. The fonts also contain data that describes the outline of each character in the typeface (or font). Higher quality fonts have a hinting code as well. Hinting is a process that makes sure the scaled character looks as smooth as possible without becoming jagged around the edges when the font is rescaled to a bigger or smaller size.

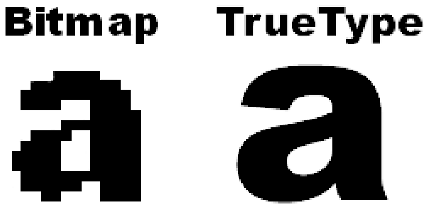


Figure 4.2: The image on the left shows a Bitmap image while the image on the right shows a True Type font, which is clearer and smoother around the edges and corners.

Computer displays and monitors, especially LCD monitors are made up of small grids or small rectangular cells known as ‘pixels’ or ‘picture elements.’ Every image or picture is made up of lots of these tiny pixel elements and the total number of pixels in an image represents the resolution of that image. Thus, a larger number of pixels represents a higher resolution and therefore, a greater image size. Today’s development of storage devices is improving rapidly, and devices can store up to a gigabyte or even a terabyte of data on their databases [61]. Figure 4.3 presents an image where a part has been zoomed in on at a high magnification, showing clearly the actual pixels that make up the image.

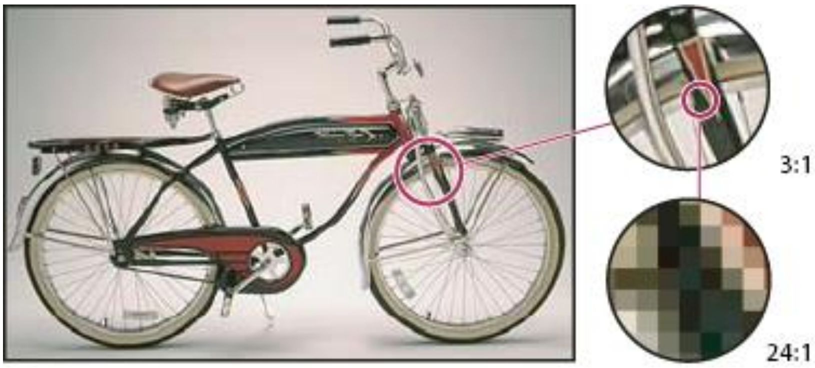
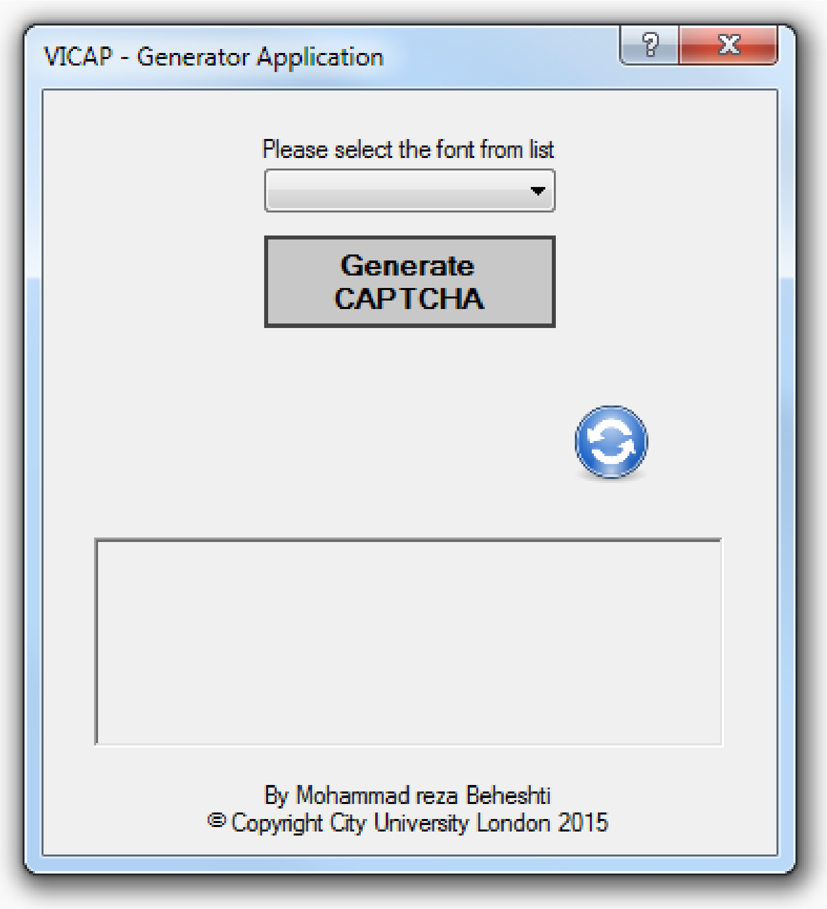


Figure 4.3: Example of a Bitmap image at various levels of magnification.

The CGA was developed to generate single VICAP frames and play them consecutively for the user. A CGA can work independently on the server side and generate CAPTCHA images automatically, as is explained by the CAPTCHA definition. A screenshot of the CGA can be seen in Figure 4.4.


VICAP - Generator Application
Please select the font from list
Generate CAPTCHA

By Mohammad reza Beheshti
© Copyright City University London 2015

Figure 4.4: Overview of the CGA application.

As it can be seen from the figure, the CGA is made up of different sections. For the convenience of the user, choices have been included to enable them to choose his or her desired font. Therefore, the VICAP test will use the user's favourite font type. This has been included to make the CAPTCHA challenge easier for the user to read. The desired font can be selected from a drop-down list menu, as shown in Figure 4.5.

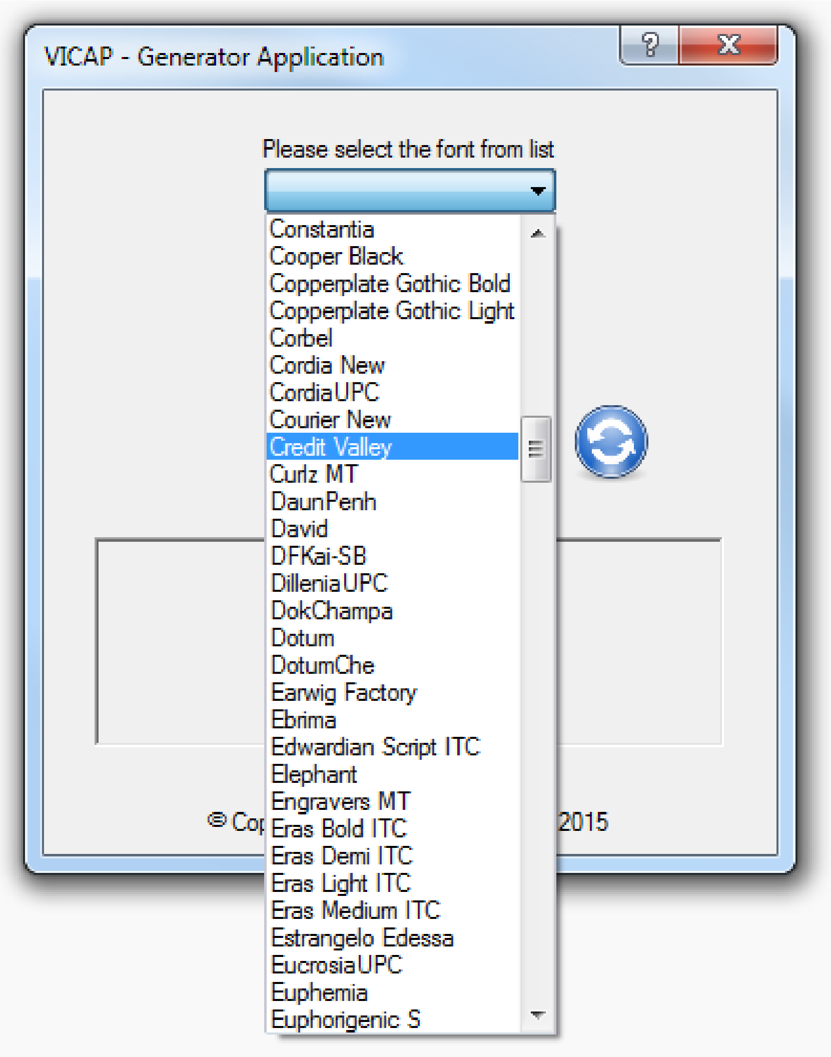
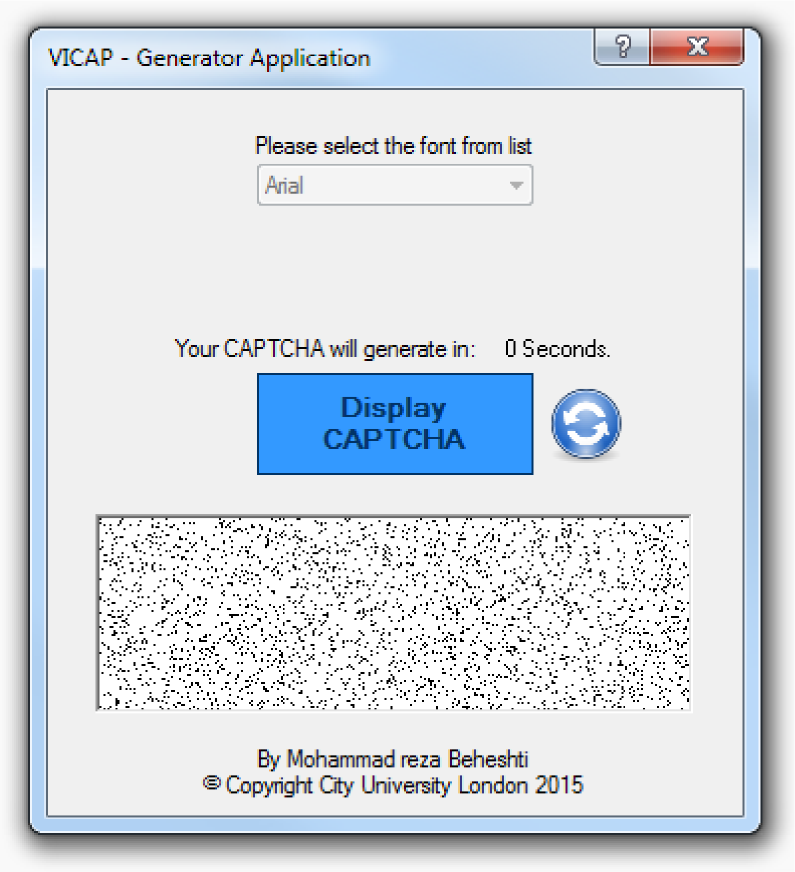


Figure 4.5: The CGA and font drop-down list menu. The user will have a choice of using desirable font type.

Another useful feature of our novel CGA is the ‘Refresh Button,’ which generates a new set of characters to the user when pressed. After completing the image-processing phase, the user will be able to click on the ‘Display CAPTCHA’ button and the CAPTCHA will then be displayed to the user in the form of a short video clip. Figure 4.6 shows a screenshot of the CGA while it is generating and displaying VICAP frames to the user.



VICAP - Generator Application

Please select the font from list

Arial

Your CAPTCHA will generate in: 0 Seconds.

By Mohammad reza Beheshti

© Copyright City University London 2015

Figure 4.6: Screenshot of the CGA in its operational phase.

To generate and render every individual VICAP frame, three main steps need to be completed before putting all the CAPTCHA images in the output stack. The procedure of generating single VICAP frame includes the following three steps:

4.3.1 Binarisation and Bitmap Conversion

Binarisation is the procedure of converting all pixel values to a binary value (0 or 1), expressed as 1 bit/pixel, which will produce a black and white image. During this process, the grey level of a pixel is compared to a threshold and is allocated the value of 0/1 if it is less/greater than the threshold, thus corresponding to black/white, respectively. It is important to convert the pixel values to their binary equivalents because this will simplify the subsequent steps to create the final CAPTCHA image. Additionally, the human eye is more sensitive to black and white stimuli, rather than coloured stimuli. This is due to the presence of *rods*, which are photoreceptors in the retina that are sensitive to shades of grey, rather than *cones*, which are sensitive to colour. Current research shows that there are approximately around 6 to 7 million cones in every healthy human eye. Cones can be divided into three parts: red cones, which have a density of 64%; green cones, which have a density of 32%; and blue cones, which have a density of only 2% [62]. The number of rods in a normal eye is approximately around 120 million, which are not sensitive to colour.

Moreover, using colour CAPTCHA images may have a negative effect on the usability and security of the CAPTCHA because it may increase the risk of CAPTCHA attacks [60]. The binarisation phase will start by declaring a threshold value and comparing every single pixel colour value against that threshold. If the pixel value falls below the threshold, the programme will turn the colour of that pixel to black and, alternatively, if the pixel value is greater than the threshold, the programme will turn the pixel colour to white. We should note that the colour of the background and foreground pixels can also be swapped, and this entirely depends on the author's design. However, in this research project, we have assumed a black colour for the foreground and a white colour for the background pixels.

After the binarisation phase, a canvas needs to be created with a size of 300×100 pixels with a white background in order to print the image of the character or number. In the proposed CAPTCHA model, it has been assumed that using a black character or number string against a white background will make the CAPTCHA easier for a human user to read. Figure 4.7 shows a randomly selected string of 'AC45R' that has been converted from a greyscale to a binary format, printed at the location (X, Y) in black characters on a white background.

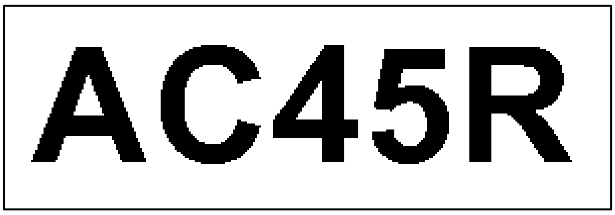


Figure 4.7: a Randomly generated string of 'AC45R,' converted to a binary format and printed at location (X, Y) against a white background.

4.3.2 Object Sampling Rate

Since the proposed VICAP model is based on the ability of the human eye to differentiate between the total number of object pixels and background noise pixels, it is important to choose the correct ratio for the OSR and background noise. In our proposed CAPTCHA model, the OSR has been chosen in a way that makes it very easy for the human eye to distinguish between the density of object pixels and background noise, but it is impossible for computer recognition programmes to distinguish between these two aspects. The sampling rate of the object pixels is random and therefore, it would be almost impossible for computer programmes to predict or learn the behaviour of the pixel elements in terms of appearing or disappearing. Figure 4.8 shows an object corresponding to character 'O,' made up of N^2 number of pixels, where X represents the number of object pixels. In this example, there is a total of $[N^2 - X]$ background pixels.

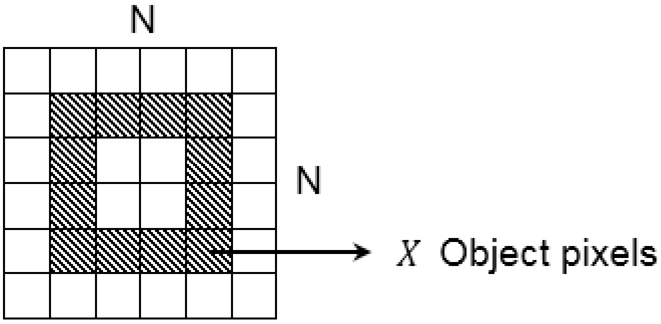


Figure 4.8: An example of a complete object corresponding to the character 'O' prior to the application of sampling.

By sampling the object at a rate of $S\%$, there will only be a partial section of the object presented to the user. Since the procedure of the sampling uses a random generator function, the presentation of the pixels varies from frame to frame. However, overall, the total number of pixels almost stays the same. For instance, as it is shown in Figure 4.9, the object 'O' has been sampled at a 50% sampling rate, which means the probability of every single pixel appearing in that frame is almost equal to 50%. Thus, on the single frame scenario there will only be half of the pixels appearing for the object 'O' and another 50% of the object pixels will not be shown at all. However, the combination of the pixels can vary from frame to frame, as shown in Figure 4.10.

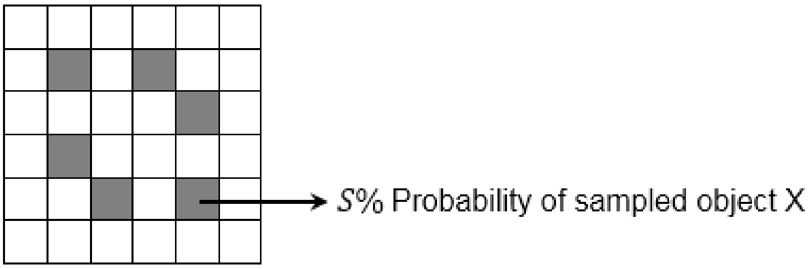


Figure 4.9: Object ‘O’ is sampled at 50%.

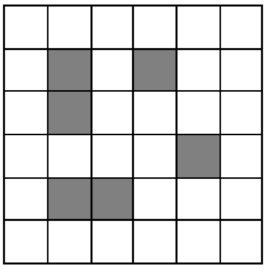
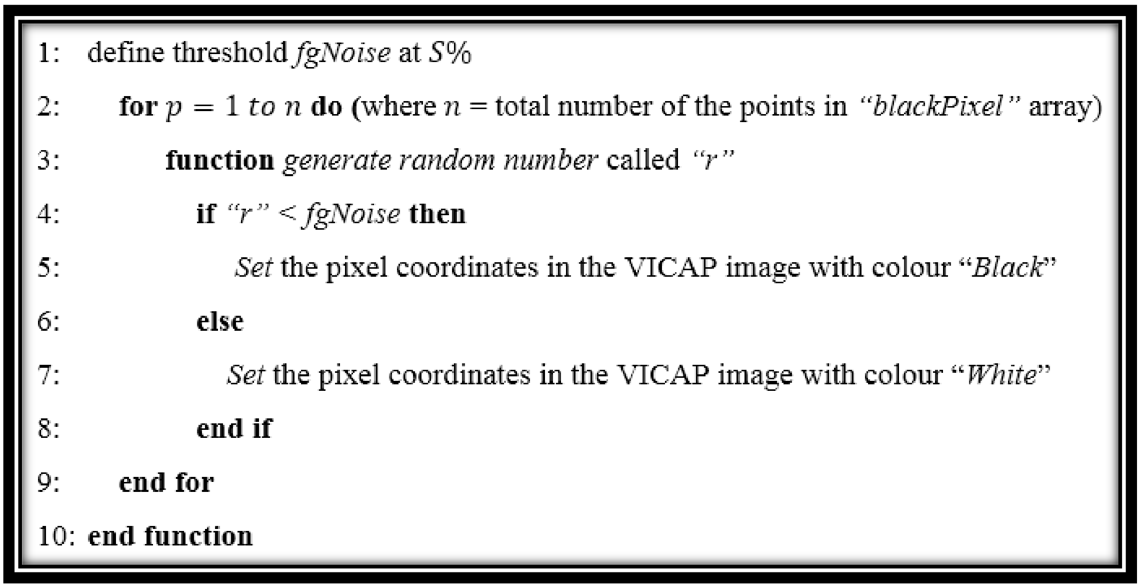


Figure 4.10: Sampling of object ‘O’ at the same sampling rate of 50%, but with a different pixel combination.

This model is based on POV and the ability of a human’s supreme visual system to hold information for a very brief period of seconds using the IM. Consequently, by presenting only a portion of the object pixels in every single frame and by playing all the frames consecutively, the HVS would be able to combine all the fleeting images and by superimposing the frames together, would be able to create the final image in the brain.

As it can be observed from the pseudo code displayed in Listing 4.1, for each element p in the array of ‘$blackPixles$’ the programme will generate a random number between 0 and 1 with the conditions explained before. It will then compare the generated random number with the pre-defined threshold ‘$fgNoise.$’ If the generated random number falls below the threshold, the programme will take that specific pixel and turn the colour to black and will then put the pixel back onto the canvas image according to its coordinates (X, Y).

Listing 4.1: Algorithm used to sample the object according to the specified threshold value.



```
define threshold fgNoise at $S\%$
    for $p = 1$ to $n$ do (where $n$ = total number of the points in "blackPixel" array)
        function generate random number called "r"
            if "r" < fgNoise then
                Set the pixel coordinates in the VICAP image with colour "Black"
            else
                Set the pixel coordinates in the VICAP image with colour "White"
            end if
    end for
end function
```

4.3.3 Adding Background Noise to the VICAP Image

The last step in creating the CAPTCHA image of the new model is adding background noise to the sampled object X. This action will make the characters of the CAPTCHA image even harder for OCR software to recognise. Since the pixels are randomly selected and presented in each single image it will, in practice, be almost impossible to predict or guess the possible combinations of the pixels to extract the final image. As shown in Figure 4.11, some level of background noise is added to the sampled object X at a percentage of $n\%$.

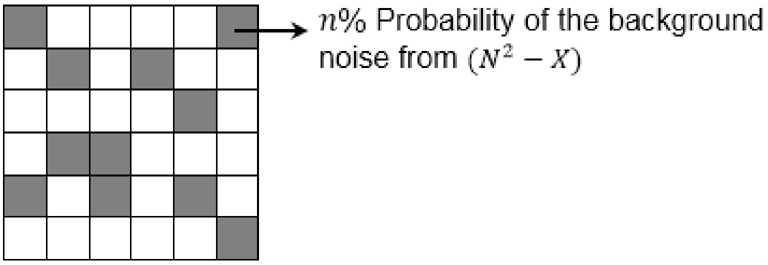
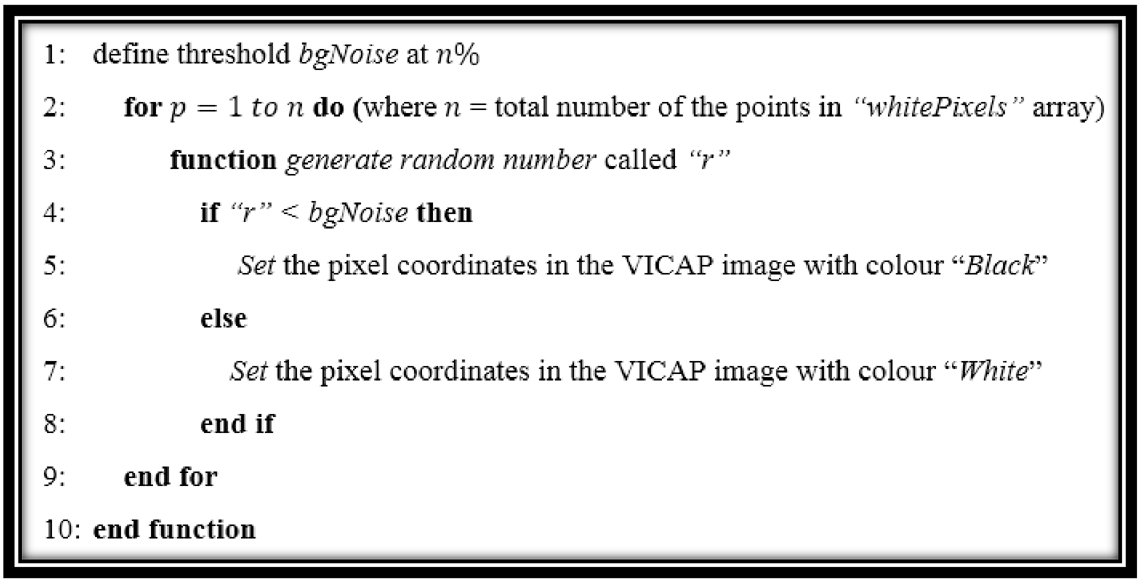


Figure 4.11: Adding background noise to the sampled object $\boldsymbol{X}$ at the percentage of $\boldsymbol{n}\%$.

Adding background noise to the image is a very similar procedure to the previous stage, the only difference being that this time the background is being sampled (by generating a random number) and compared to its threshold value, known as ‘*bgNoise*.’ The same algorithm will then be applied to every single element of the background pixel array called ‘$whitePixels$’. Similarly, the algorithm to create the random background noise will be very similar to the algorithm of sampling objects, however in this case it will replace the ‘*BlackPixels*’ with ‘*WhitePixels*,’ which represent the background pixels. Thus, a new threshold value called $n\%$ needs to be declared for the background noise (called ‘*bgNoise’*), which has a threshold value that all random generated numbers will be compared against. As it can be observed from the pseudo-code presented in Listing 4.2, the iteration goes through every single element of the ‘*WhitePixels*,’ with the coordinates (X, Y) of all the background pixels, by comparing the generated random numbers with the threshold value. The programme will randomly filter the background pixels at a specific rate.

Listing 4.2: Algorithm used to add background noise to the sampled object according to the defined threshold value.



```
define threshold bgNoise at $n\%$
    for $p = 1$ to $n$ do (where $n$ = total number of the points in "whitePixels" array)
        function generate random number called "r"
            if "r" < bgNoise then
                Set the pixel coordinates in the VICAP image with colour "Black"
            else
                Set the pixel coordinates in the VICAP image with colour "White"
            end if
    end for
end function
```

Figure 4.12 shows a simple flowchart sketch that represents the process of identifying each pixel value and using these to then apply the sampling rate or add background noise to the CAPTCHA image. As it can be seen from the flowchart, the process starts by identifying every individual pixel value by scanning the entire CAPTCHA image row by row. Then, after separating the pixels that belong to the object from the background pixels, the programme will then generate a random number and compare it with the given threshold value. If the generated random value falls below the predefined threshold, the programme will change the colour of that pixel. The process of generating a random number and comparing it to the given threshold value will be repeated for both the object pixels and the background pixels until all the pixels in a single CAPTCHA image have been processed. In this way, a single VICAP frame will form five random letters, which have been sampled at a percentage of $S\%$ and background noise added at a percentage of $n\%$. Using this method, the programme will generate a number of VICAP frames. After this, the whole sequence will be played for the user at very high speed, enabling them to produce a final image using POV. The total number of frames playing per second depends on the delay time 'd' between each consecutive frame. As the value of the delay gets shorter, there will be a larger number of fleeting frames per second. For instance, for any movie clip to be seen continuously and smoothly without flashing images, it should be played with at least 40 milliseconds of delay between each consecutive frame, which is almost equivalent to 24 FPS. Any delay shorter than 40 milliseconds will produce a larger number of fleeting FPS, which means that the motion will look smoother in our visual system. As discussed previously, the proposed CAPTCHA model utilises the ability of the HVS to retain every image after it passes before our eyes for a very short fraction of a second after it disappears. In every single frame, only a percentage of the object pixels are shown. As these pass before the user's eyes quickly, the user's visual system would be able to build up the final image itself, without giving away the full information of the object.

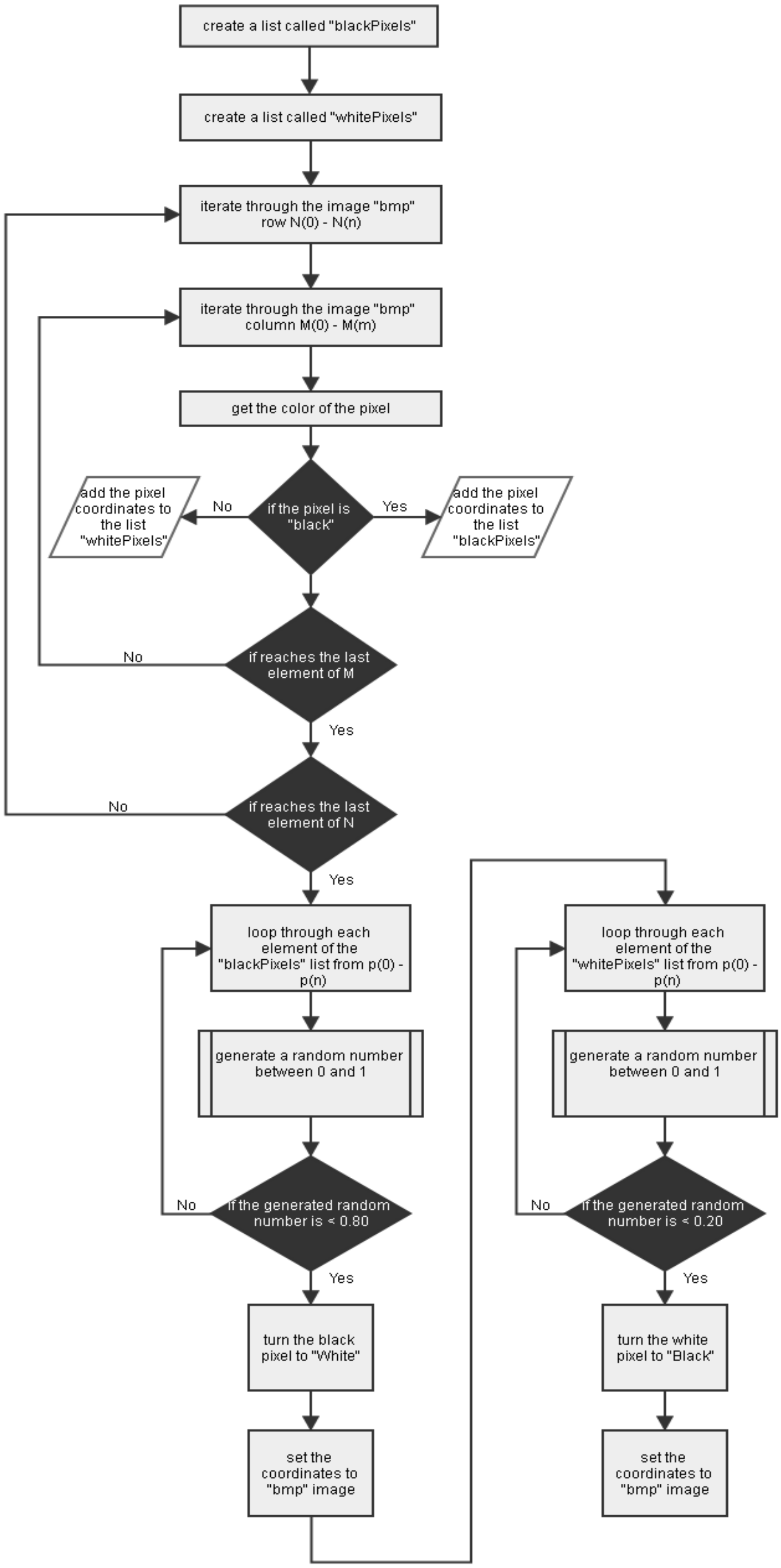

create a list called "blackPixels"
create a list called "whitePixels"
iterate through the image "bmp" row N(0) - N(n)
iterate through the image "bmp" column M(0) - M(m)
get the color of the pixel
if the pixel is "black"
No
add the pixel coordinates to the list "whitePixels"
Yes
add the pixel coordinates to the list "blackPixels"
if reaches the last element of M
No
Yes
if reaches the last element of N
No
Yes
loop through each element of the "blackPixels" list from p(0) - p(n)
generate a random number between 0 and 1
if the generated random number is < 0.80
No
Yes
turn the black pixel to "White"
set the coordinates to "bmp" image
loop through each element of the "whitePixels" list from p(0) - p(n)
generate a random number between 0 and 1
if the generated random number is < 0.20
No
Yes
turn the white pixel to "Black"
set the coordinates to "bmp" image

Figure 4.12: The flowchart represents the process of generating a sampled object and injecting background noise to the VICAP image.

Figure 4.13 shows an example of the three steps involved in generating a single VICAP frame. Step-1 is producing the binary image of the original data. Step-2 is the sampling of the object (for instance at a rate of 25% in this example), and Step-3 is injecting background noise (at a rate of 15% in this example). After generating the VICAP frames the next step is to present the CAPTCHA images at an appropriate animation speed for human users to perceive (using POV). The CGA is used to render individual images with the required specifications, such as background noise and OSRs, and subsequently play back the CAPTCHA sequence for the user.

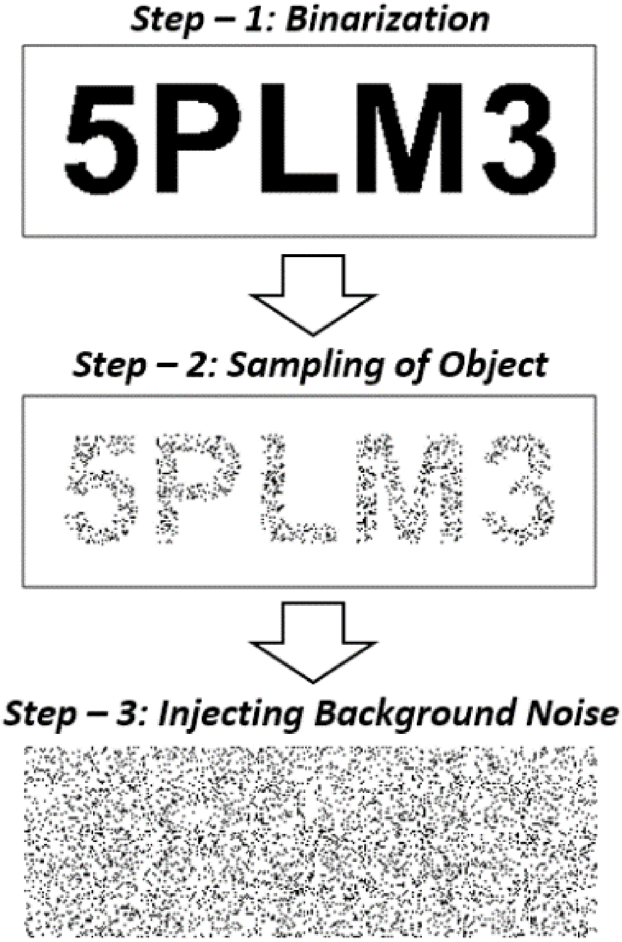

Step – 1: Binarization
5PLM3
Step – 2: Sampling of Object
5PLM3
Step – 3: Injecting Background Noise

Figure 4.13: Steps involved in generating a single VICAP frame.

Figure 4.14 presents examples of 10 different output frames generated by the CGA. Due to space limitations, we are not able to show the final output effect. This should be experienced in the real world. Interested readers may experience the proposed model on the CAPTCHA evaluation and user experience website at *http://mrbeheshti2.wixsite.com/captcha-project*

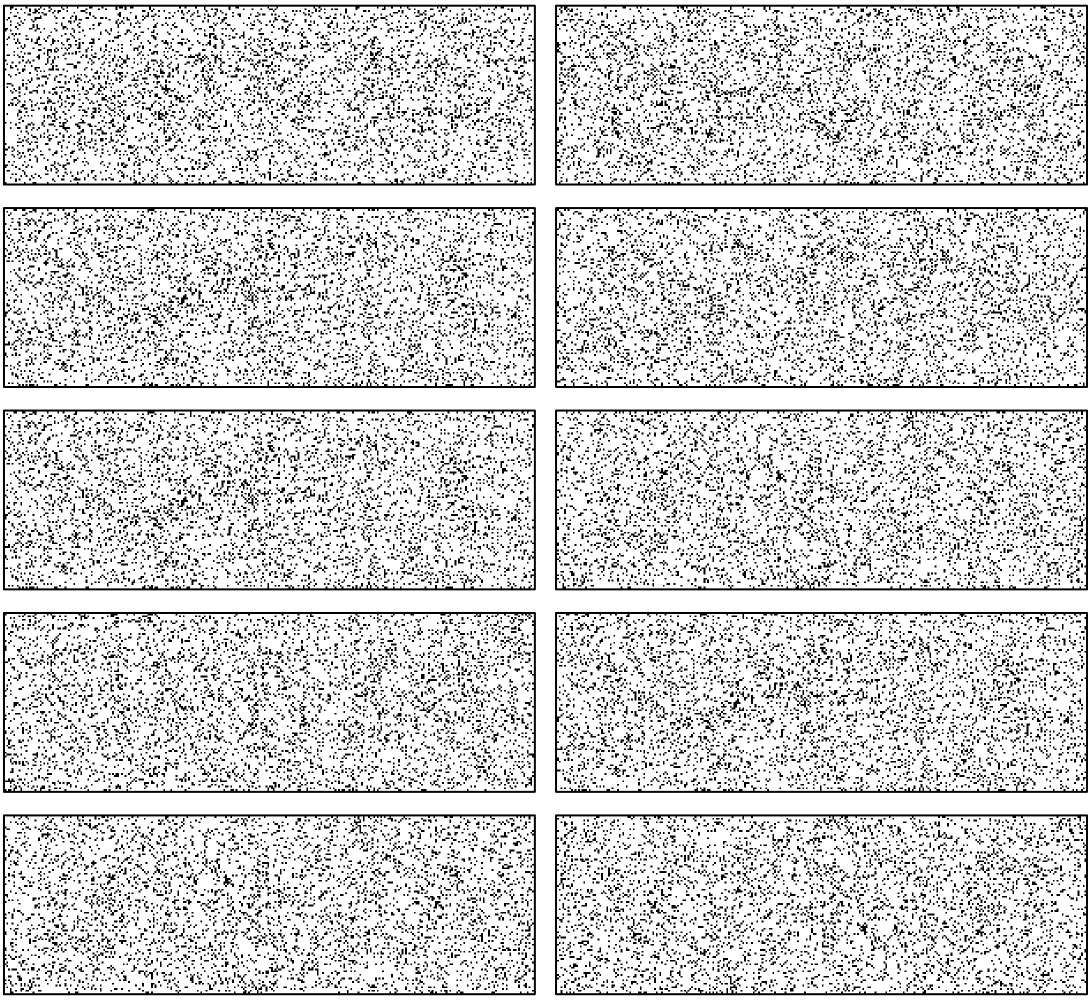


Figure 4.14: Example of 10 different output frames generated and rendered by the CGA with a background noise of 15% and an OSR of 25%.

From previous explanations, it can be concluded that one of the most crucial factors that make the phenomenon of POV happen is the relationship between the sampled object and the background noise. Every single frame (or image) is made up of N^2 number of pixels in total, and the object is made up of a number of pixels that is represented by X. Therefore, there will be $(N^2 - X)$ total number of background pixels. To distinguish between the sampled object and the background noise, there should be a difference in their ratios. The following expression can be used for every single frame scenario:

$$\frac{n}{(N^2 - X)} < \frac{S}{X} \tag{4.1}$$

Here, $n/(N^2 - X)$ represents the BNR and S/X represents the OSR. This mathematical expression could also be reversed as follows:

$$\frac{n}{(N^2 - X)} > \frac{S}{X} \tag{4.2}$$

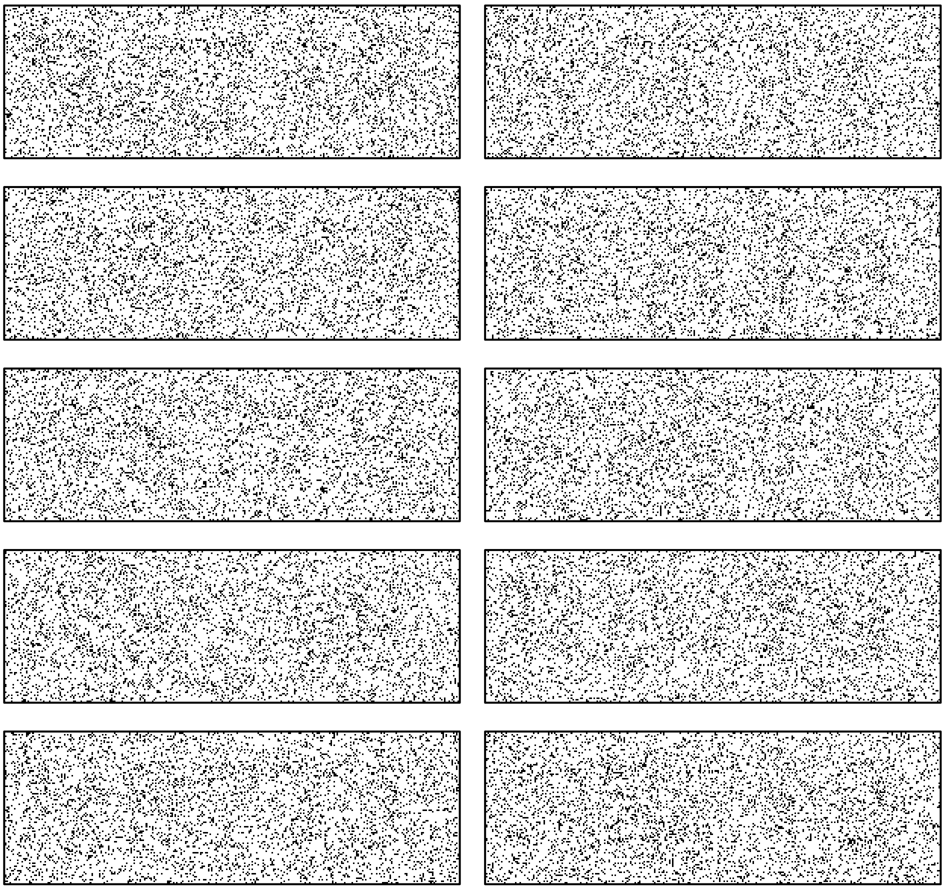
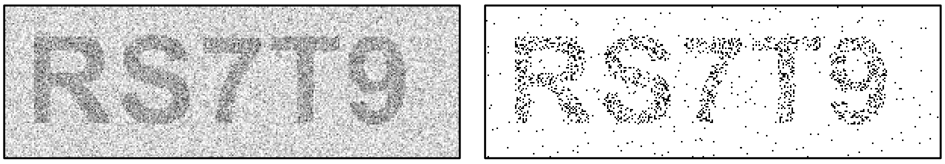
As it has been presented in 4.2, the proportion of background noise is greater than the proportion of the OSR, but in practice, this might not be possible. In this example, since the object is presented as the background noise, it will be possible to be analysed and processed individually. Thus, having a greater background noise ratio than OSR will cause our model to be seen in a negative format. This means that the background would be seen as darker than the object, whilst in the normal situation, the object would be seen by the user to be darker. There is also an exceptional situation whereby the OSR is exactly equal to the ratio of background noise. In this exceptional circumstance, the probability of any pixels appearing is exactly equal in both object and background noise; therefore, in practice, the following expression (4.3) will be invalid because it will not produce any visible results, apart from generating pure noise:

$$\frac{n}{(N^2 - X)} = \frac{S}{X} \tag{4.3}$$

From the above equations, it must be noted that the proportion of the sampled object to the original object pixels should be much higher than the proportion of the background noise to the background pixels and vice versa. Thus, if all consecutive frames are fleeted, after a certain number of frames, the following results will be achieved, as shown in Table 4.1. The graphical output results here have been produced using another state-of-the-art application called CAPTCHA-Test Application (CTA), a process that will be discussed in more depth in the next section. To justify the above equations, three different scenarios have been investigated to demonstrate the impact of the ratio between the OSR and the level of background noise on the HVS.

SCENARIO-1: In this scenario, the CTA will be set up to generate CAPTCHA frames with a 30% OSR and 15% of the background noise injected. After playing 10 consecutive frames, all the results will pass to the second CTA, which will then analyse and evaluate the output results. As is noticeable from Table 4.1, when the OSR is greater than the background noise, the object should appear to be a darker colour than the background. Table 4.1 shows 10 generated VICAP frames and the final analysed output results in 8-bit/pixel in greyscale as well as 1-bit/pixel in a black and white format.

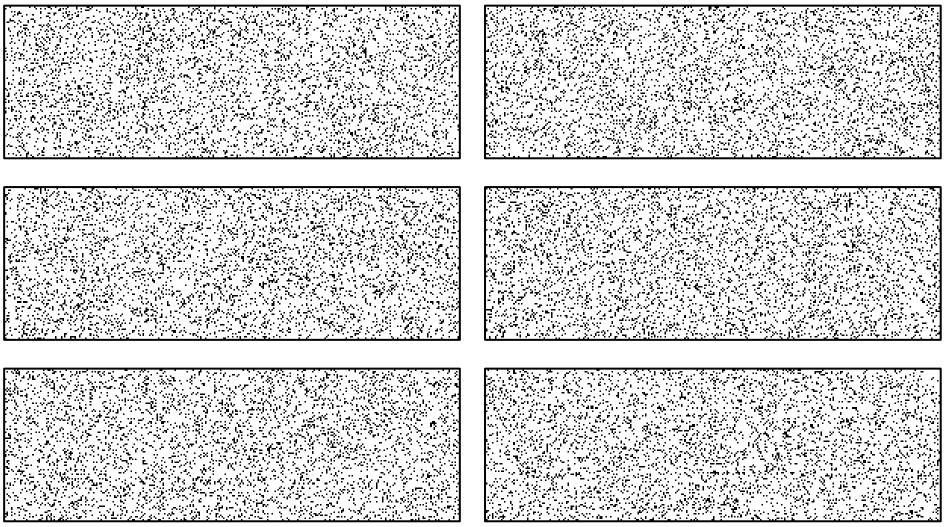
Table 4.1: The generation of 10 VICAP frames with a 30% OSR and 15% background noise, followed by the final output rendered images in 8-bit/pixel and 1-bit/pixel formats.

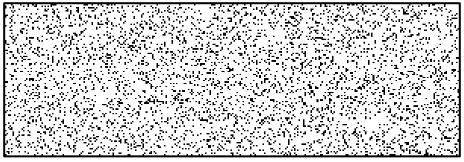
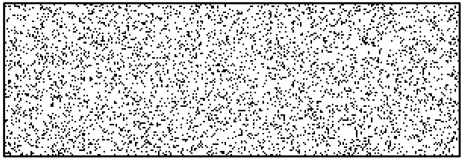
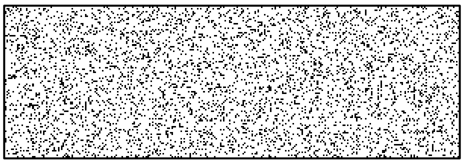
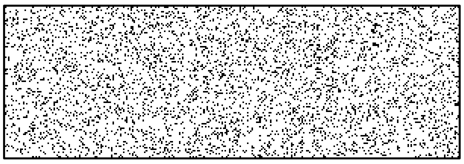
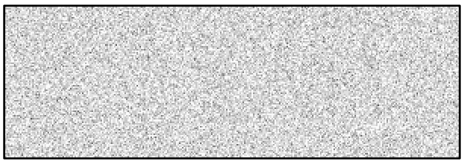
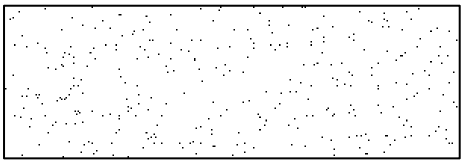
VICAP frames (10 images)	
Output results 8-bit/Pixel Greyscale & 1-bit/Pixel	

As the output results from Table 4.1 show, after rendering and processing 10 individual VICAP frames, the string of letters ‘RS7T9’ is easy to recognise from a human’s perspective, but each frame does not reveal any visible information that could be meaningful to a bot. Another aspect that should be noted from this table of results is that the character string ‘RS7T9’ is printed in a darker shade than the background noise due to the difference in the ratios of the OSR and the background noise. Furthermore, in the 1-bit per pixel format, the region that belongs to the object appears more congested and has a higher density of pixels than the background areas.

SCENARIO-2: Like the first scenario, the second scenario will be set up to generate 10 different CAPTCHA images using the CGA. However, in this experiment, the same ratio will be offered for both the OSR and the background noise, which have both been adjusted to 15%. As it can be observed from the output results in Table 4.2, because the OSR is exactly equal to the background noise, the probability that the background pixels will appear will be exactly the same as the probability that the object pixels will appear. For that reason, every single frame will be made up of only pure noise and thus, the final output image will be also be made up of pure random noise. Table 4.2 shows the generation of the 10 different VICAP frames and the analysed output image in 8-bit/pixel as well as 1-bit/pixel formats.

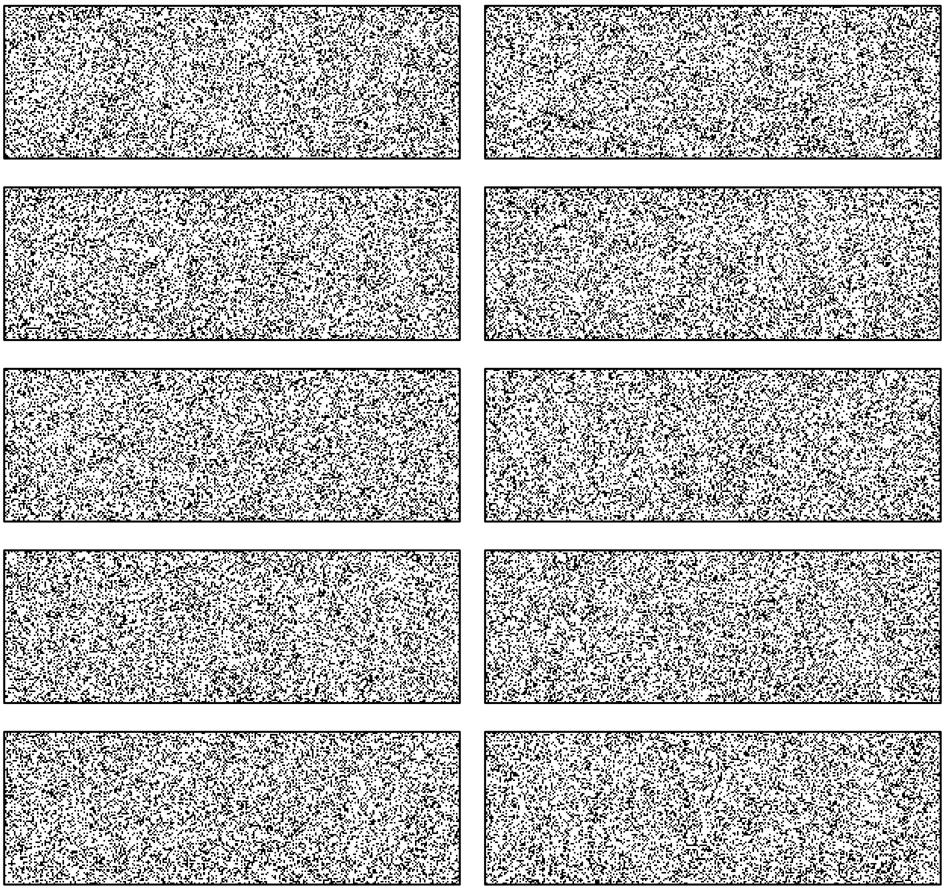
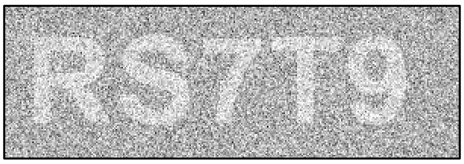
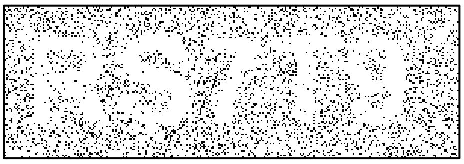
Table 4.2: The generation of 10 CAPTCHA frames with the same ratio for the OSR and the background noise, both 15%, followed by the final rendered output images in 8-bit/pixel & 1-bit/pixel formats.

VICAP frames (10 images)	

	   
Output results 8-bit/Pixel Greyscale & 1-bit/Pixel	 

SCENARIO-3: In this last scenario, the same experiment will be repeated, but this time the proportion of background noise will be greater than the proportion of the OSR. The expected result of this scenario is that the object will be resolved and disappear into the background noise because the amount of background noise is greater than the OSR. However, in our proposed model, as the object is separate from the background noise, making the amount of background noise greater than the OSR would make the object appear a lighter shade of grey than the background noise. Table 4.3 presents 10 VICAP frames where the OSR is set to 15% and the background noise is set to 30%. As in the other scenarios, the final analysed rendered output images in 8-bit/pixel and 1-bit/pixel formats have also been presented.

Table 4.3: The generation of 10 CAPTCHA frames with an OSR of 15%, and a background noise of 30%, followed by the final output rendered images in 8-bit/pixel & 1-bit/pixel formats.

VICAP frames (10 images)	
Output results 8-bit/Pixel Greyscale & 1-bit/Pixel	 

As it can be observed from Table 4.3, the output results in both 8-bit/pixel greyscale and 1-bit per pixel black and white formats would both be understandable to a human user. The string of characters ‘RS7T9’ is clearly noticeable. However, the object is a lighter shade and is less intense compared to the background noise, which has a higher density of pixels. In conclusion, it can be noted that, in the scenarios where the OSR is either greater or smaller than the background noise (rather than the same as in Scenario 2), the output results are the most noticeable to the human eye. If the same ratios are used, as in Scenario 2, an image will be produced that only contains pure noise with no readable information available.

4.4 Development of the CAPTCHA-Test Application

Since our new proposed CAPTCHA model is based on POV and a human's IM, it will be meaningless to any current computer recognition programmes. By analysing and testing individual frames, CRSRs will be almost equal to 0%. The CTA is a state-of-the-art application that was designed to test our novel CAPTCHA model. This application was developed to test and analyse output images in a similar way to the human eye in order to measure the robustness level of the model.

4.4.1 CAPTCHA-Test Application Overview

Figure 4.15 presents a screenshot of the CTA. This application has been developed using a .NET framework and C# programming language. As it has been shown in Figure 4.15, the interface of the application is made up of two sections. Section 1 is a simple button called 'Load Images,' a button that triggers a large and complex process. After pressing this button, which is the backbone of the system, the programme will first try to locate all the CAPTCHA images that were previously saved by the CGA. After that, the programme will go through the image-processing phase by calculating all the pixel density values for every single frame, and then present the values in different tables. More explanation about methodology and how to calculate the output images based on their number of frames will be provided in due course.

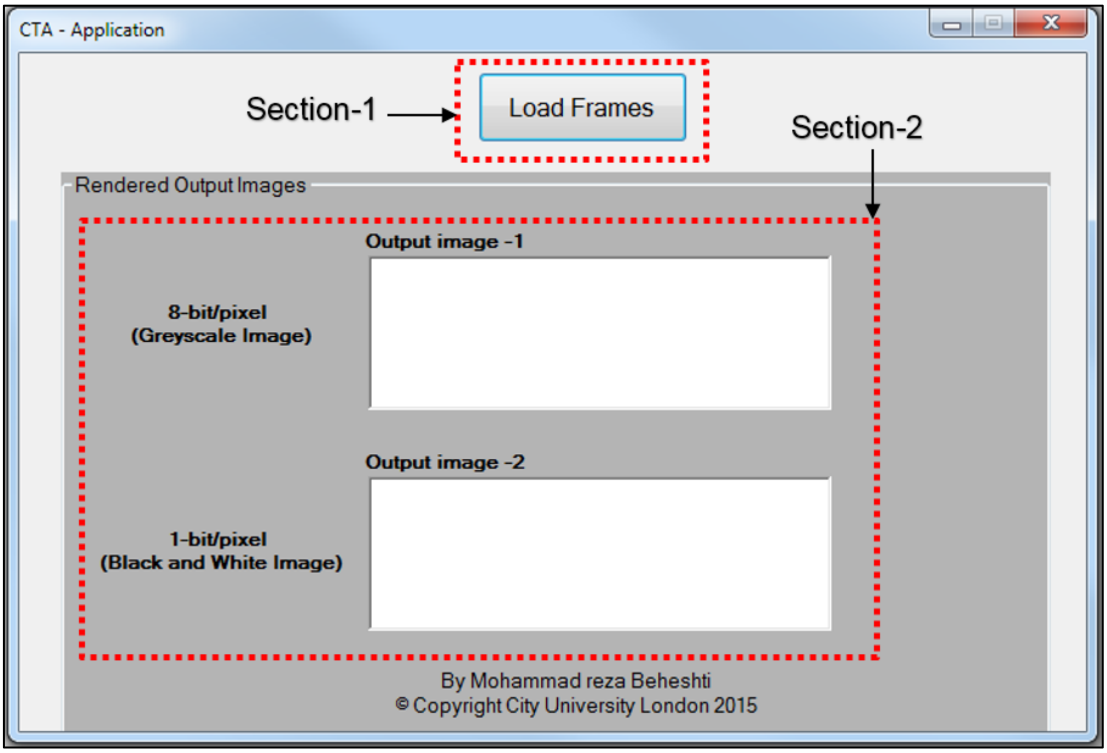


Figure 4.15: General overview of the CTA application. As the picture shows, the application is made of two sections. By clicking on the "Load Frames" button, the process of integration will start.

Section 2 presents an output results window that provides two different outputs based on different mathematical calculations. Output image-1 is a rendered image based on the 8-bit per pixel, which produces an image in greyscale format. The output image-2 is an output rendered image based on 1-bit per pixel, which produces a black and white image. Figure 4.16 shows a screenshot of the CTA in the operational mode. As it can be seen from the picture, two different graphical representations of the pixel density are presented. As the total number of analysed frames increases, the output results will make a clearer image. Since our proposed CAPTCHA model utilises the sophisticated HVS, it is believed that it would not be possible for any current computer recognition software to analyse and decipher any of the individual frames meaningfully. Our extensive laboratory experiments confirm this and these results will be presented shortly. To test our proposed CAPTCHA model, we wanted to reform the frames and superimpose them first before feeding them into any OCR recognition software. Figure 4.17 shows the process of generating VICAP images using the CGA and storing the output frames on the database, to be used for the CTA analysis later.

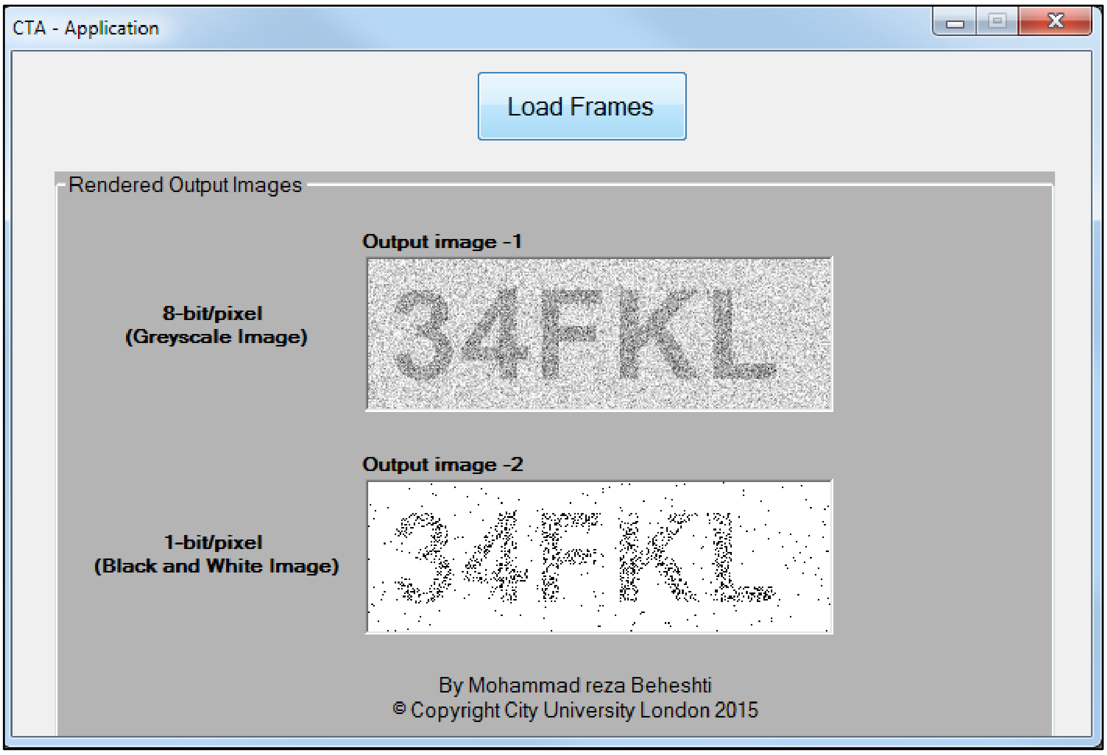


Figure 4.16: CTA application in operational mode. Two graphical representation of output images are presenting, 8-bit/pixel grayscale and 1-bit/pixel mono colour format.

As it can be observed from Figure 4.17, the CGA has been used to generate and render a series of VICAP images, which will be saved into the computer system, then fed into the CTA. The idea of developing the CTA application was to test the robustness of the VICAP frames generated by the CGA and then to simulate the final output results based on the same principals as the human eye. The final output results can be separated into two different formats, as is explained in the next section.

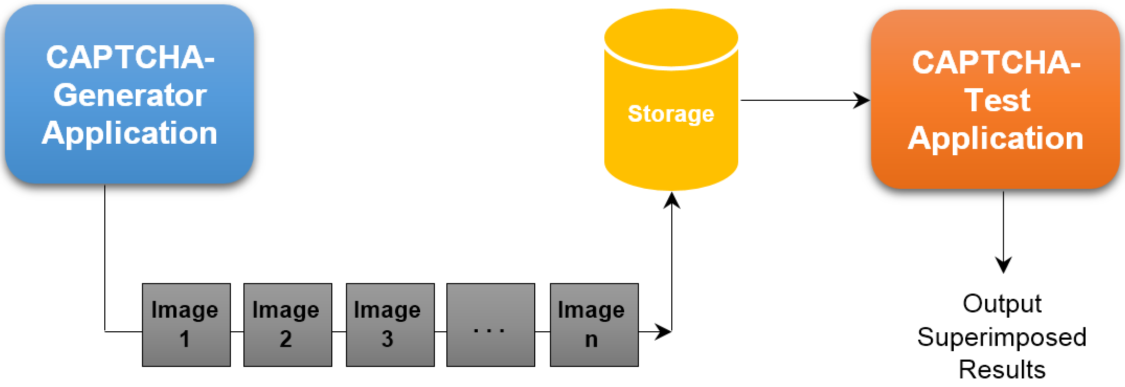


Figure 4.17: The procedure of generating VICAP frames using the CGA and feeding them into the CTA to be superimposed. The output results will be an 8-bit/pixel greyscale image as well as a 1-bit/pixel mono-colour image.

4.4.2 8-bit/pixel, Greyscale Format Output Image

As has been explained, every single image is made up of very tiny elements called pixels. In the other words, pixels are the smallest elements that can form an image. Each pixel depends on the different number of bits it contains, and every pixel can have different shades of colour, represented by the number of bits per pixel. For instance, our new CAPTCHA model is made up of 8 bits per pixel, which can include 0 to 256 different shades of grey, as shown in Figure 4.18.

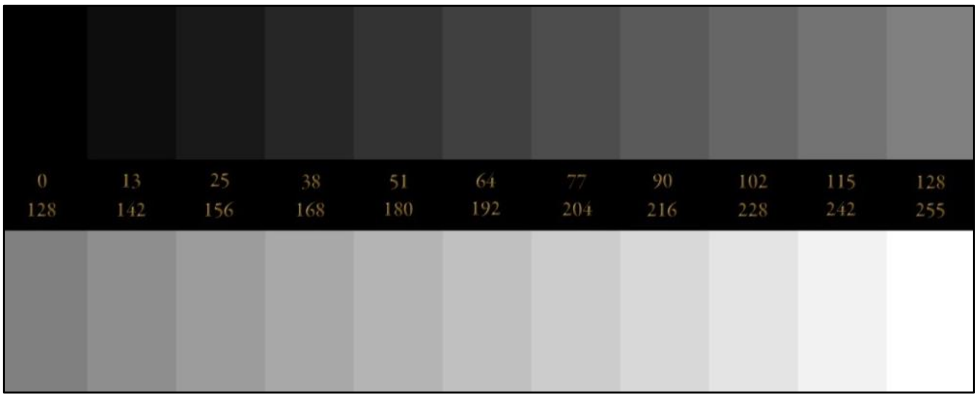


Figure 4.18: The 256 different shades in the greyscale format image.

As mentioned earlier, having 8 bits per pixels will make it possible to have 256 different shades of grey, ranging from 0, which represents the Least Significant Bit (LSB) and is often associated with black, and 255, which represents the Most Significant Bit (MSB), and it is often associated with white.

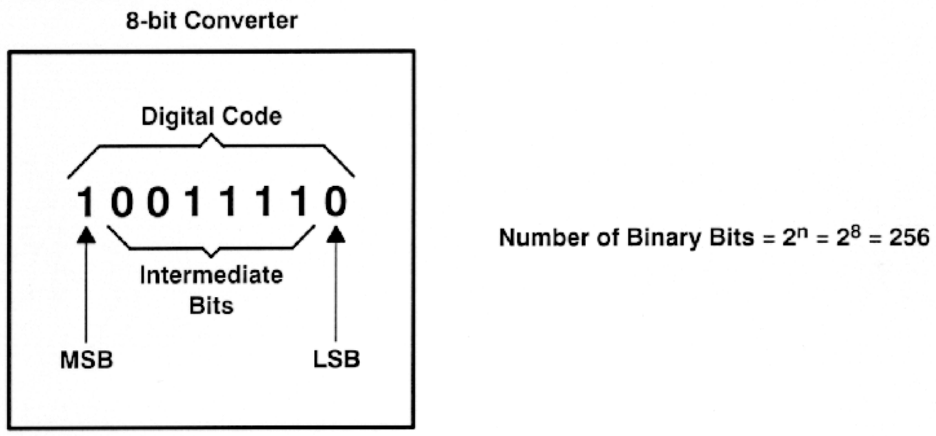


Figure 4.19: 8-bit conversion according to the MSB and the LSB.

To better understand this process, the OSR has been adjusted to 20%. Therefore, theoretically, having five frames should be sufficient enough to be able to retrieve almost 100% of the original information of the object, which can then be used to form the original object (assuming that in each frame there is exactly 20% of the object pixels presented randomly). The random function has been set up to sample the object using a pseudo-random technique, making the probability of each pixel turning black or white only 20%. Due to this, it cannot be guaranteed that after exactly five images it will be possible to retrieve 100% of the information. For that reason, it can be concluded that the larger the number of frames analysed, the more accurate and clearer the results will be.

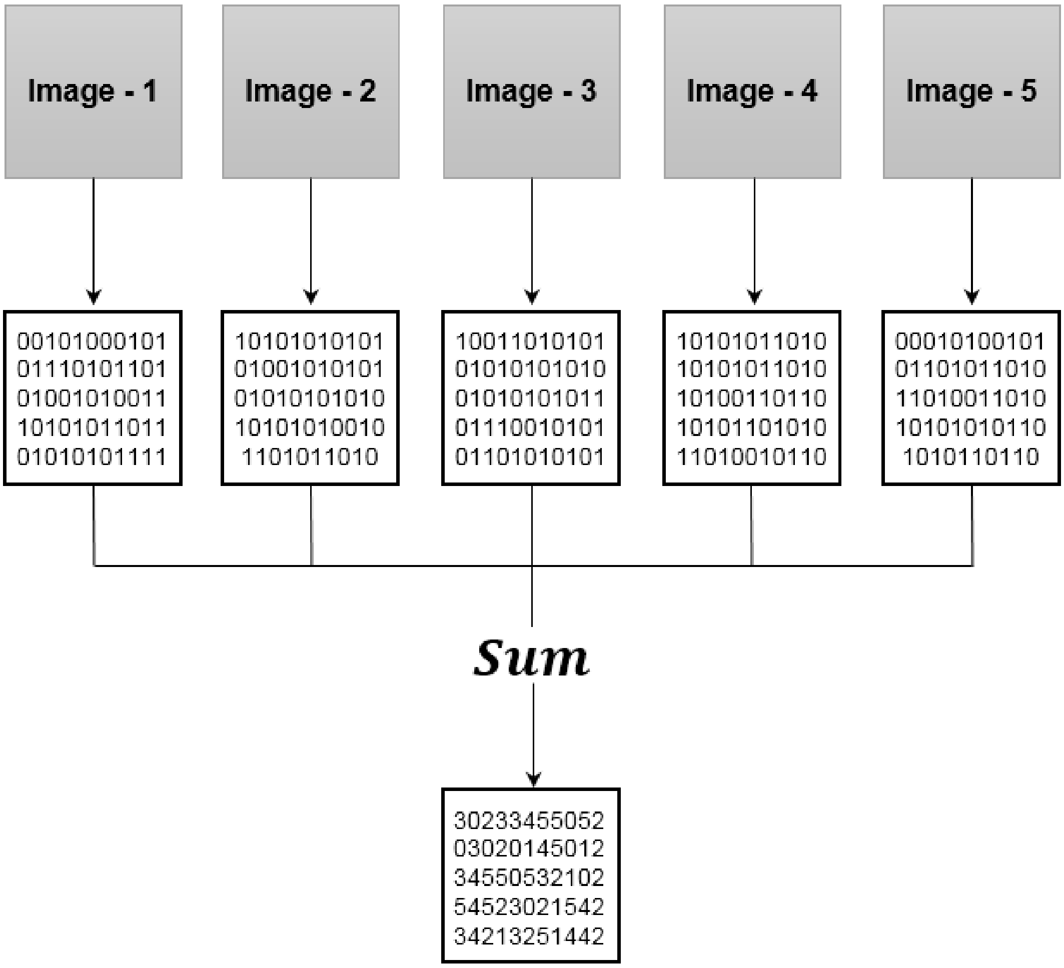


Figure 4.20: The summation process of binary values for 5 different frames. As the graph shows, the “Sum” list has all the average value from 0 to 5. Where 0 represent white and 5 represent black colour. All other value in between will represent different shades of grey accordingly.

As it has been shown in Figure 4.20, all the input CAPTCHA images need to first be converted into their binary values, which will be represented by 0s and 1s. After that, those results will be saved in five separate lists. The process of conversion is done by scanning the entire image row by row and column by column from left to right. All images generated by the CGA are in the form of 1-bit per pixel and therefore, they can only be presented in black and white (0 representing black pixels and 1 representing white pixels). Following this, all the pixel values from the same location (or the same element of the matrix), will be merged together and the conclusive results will be presented in the new list called ‘Sum,’ which is made up of values from 0 to 5. 0 means that pixels in all previous frames were turned to black and 5 means that pixels in all previous frames were turned to white. There will also be some other pixel values in between those, which will be distributed accordingly. Table 4.4 clearly presents binary values associated with every individual pixel for 5 different frames, as well as their sum values presented from 0 to 5.

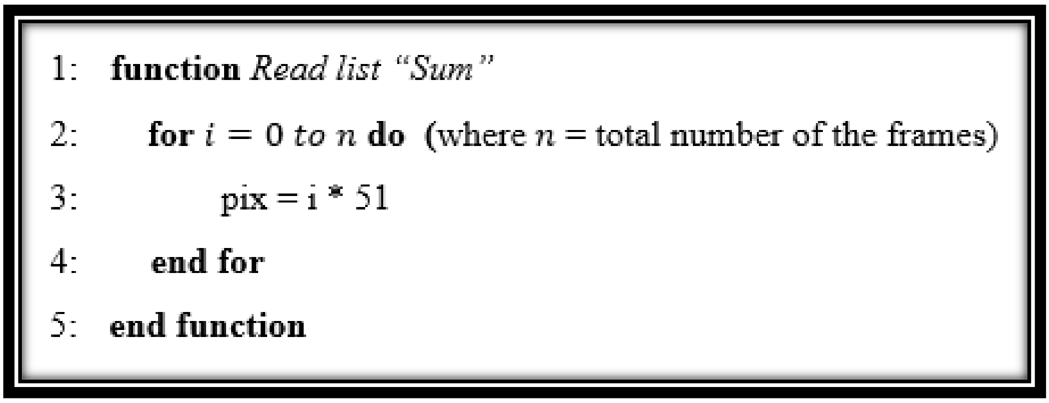
Table 4.4: Representation of the pixel binary values for 5 frames and the summation value of all the pixels.

No. of Pixels	Image-1	Image-2	Image-3	Image-4	Image-5	Sum
1	1	0	1	1	1	4
2	1	1	1	1	1	5
3	1	1	1	1	1	5
4	1	1	1	0	1	4
5	0	1	1	1	1	4
6	1	1	1	1	1	5
7	1	0	1	1	1	4
8	1	1	1	1	1	5
9	1	1	1	1	1	5
10	1	1	1	1	1	5
11	1	1	1	1	1	5
12	1	1	1	1	1	5
13	1	1	1	1	0	4
14	1	1	1	1	1	5
15	1	1	1	1	1	5
16	1	1	1	1	1	5
17	1	1	1	1	1	5
18	1	1	1	1	1	5
19	1	1	1	1	1	5

20	1	1	1	1	1	5
…	…	…	…	…	…	…
9999	1	1	1	0	1	4
10000	0	1	1	1	1	4

Here, five different frames have been selected. All possible shades of grey, from 0 to 255, need to be divided into five various levels that each represent different grey colours. Each pixel can only have a value of 0 or 1 and therefore, the higher the number of 1s received for any pixel, the more likely it is to be white and vice versa. In other words, if the sum value for each pixel is closer to 5, there will be more possibility that the pixel is white and belongs to the background, rather than the object. The algorithm below is a pseudo-code that shows how to divide pixel values from 0 to 255 into 5 various levels and categorise each level to specific shades of grey, as explained in Listing 4.3.

Listing 4.3: Algorithm to convert 5 different shades of grey according to the summation of 5-pixel values.



```
function Read list "Sum"
    for $i = 0$ to $n$ do  (where $n$ = total number of the frames)
        pix = i * 51
    end for
end function
```

Figure 4.21 shows the superimposition output results for 5 consecutive VICAP frames that have been rendered and represented using 8-bit/pixel formatting. Here, the character string ‘KY3BE’ can be clearly retrieved by the CTA, and this can then be fed into an OCR programme.

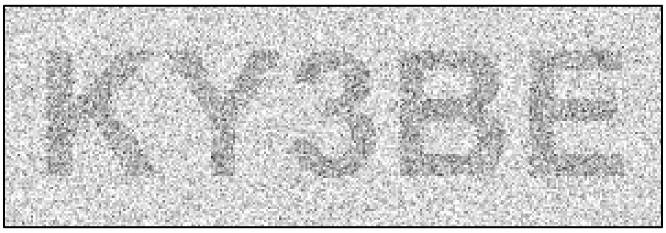


Figure 4.21: The final superimposition output results per 8-bit/pixel in greyscale format.

4.4.3 1-bit/pixel, Black and White Output Image

On the other side of the outputs window, diverse types of simulated results can be seen, which are based on the 1-bit per pixel black and white mode. The way it works is by calculating the number of times that pixels turn black or white. The system will then calculate the average value of the votes per pixel for all the frames by adding the total number 1s, each 1 representing one vote. Then it will divide that value by the total number of frames and round up the value. It will end up with another matrix, which only holds values of 0s and 1s. The new results can only represent those pixels that have been turned black and are assumed as object pixels. It is expected that object pixels turn black more often than background pixels. Figure 4.22 shows both an 8-bit/pixel image in greyscale mode and a 1-bit/pixel black and white dotted output image.

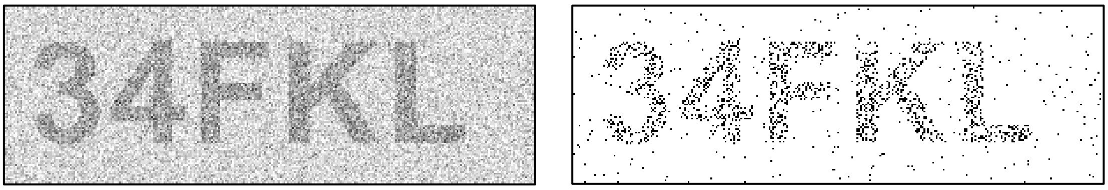


Figure 4.22: A representation of an output rendered image using 8-bit/pixel greyscale format on the left side, and 1-bit/pixel black and white format on the right side.

4.5 VICAP v.1 Experimental Simulation Output Results

In this section, our aim is to produce a comprehensive number of experimental output results using our state-of-the-art CGA and CTA to understand and measure the level of vulnerability and robustness of our proposed CAPTCHA model. Since the proposed CAPTCHA model is based on POV, the effect of the superimposed results would only be possible for a human eye to see. Consequently, it will not be possible to produce all the actual seen results in this thesis. Thus, the parameters and values that have been selected for the results section have been chosen on the basis that they are understandable and recognisable in print form. In this case, three different scenarios have been suggested using different experimental conditions. Scenarios 1 and 2 demonstrate the impact of having a different number of frames on both the HVS and the computer recognition software. In the first experiment, 5 frames have been presented, and in the second experiment, 10 frames have been presented. It is prudent to compare the output results of these two experiments. In Scenario 3, we have compared a substantial number of sampled objects with a variety of OSRs and BNRs in order to compare and analyse their impact on the CRSR.

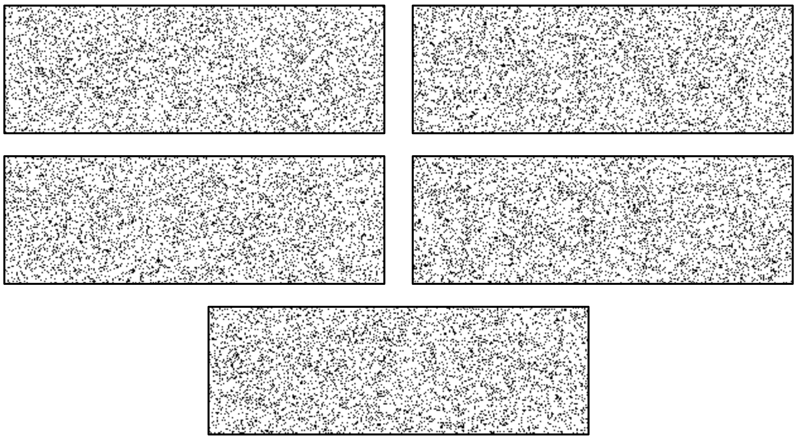
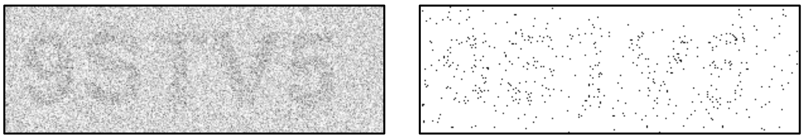
4.5.1 5 Frames Superimposed Simulation Results

In this experiment, our aim is to analyse the impact of the number of frames on the CRSR. In this scenario, 5 consecutive VICAP frames have been generated and rendered using the CGA, and the aim is to superimpose all of these frames in order to understand their impact on the CRSR. In this scenario, all the pixel values have been presented as 8-bit/pixel and thus, the total number of binary values (which are 256 bits) needs to be divided by 5 in order to be distributed equally over all possible shades of greyscale colour. Figure 4.23 shows the different shades of greyscale colour.

Figure 4.23: A representation of the 256 binary bits divided into different shades in the greyscale format.

As mentioned earlier, all the CAPTCHA images in this experiment have been generated and rendered using the CGA, and all of them have been adjusted to have an OSR of 25% and a BNR of 15%. The superimposed output images have been generated using the CTA using a 1-bit/pixel for black and white images and an 8-bit/pixel for greyscale images. The results of these generated CAPTCHA images and the final output results analysed by the computer have been shown in Table 4.5.

Table 4.5: The generation of 5 CAPTCHA frames with an OSR of 25% and a BNR of 15%, followed by the superimposed output images in an 8-bit/pixel greyscale format and a 1-bit/pixel black and white image.

VICAP Frames (5 images)	
Output Superimposed results 8-bit/Pixel greyscale & 1-bit/Pixel Black and White	

As it can be seen here, by only looking at every single one of the CAPTCHA images, no meaningful information can be perceived by either computer recognition software or the human eye. However, after playing those images to the human user or a computer programme, the above output results from Table 4.5 can be observed. As it can be noticed from our experiment, the output superimposed results are showing the character string of ‘9STV5’ in 8-bit/pixel greyscale format with the clarity of around 35% (±5%) for the human user, while the CRSR would be around 15%. However, it should be noted that the visibility and quality of the output results could also depend on other factors, such as environment (e.g. brightness of the monitor).

4.5.2 10 Frames Superimposed Simulation Results

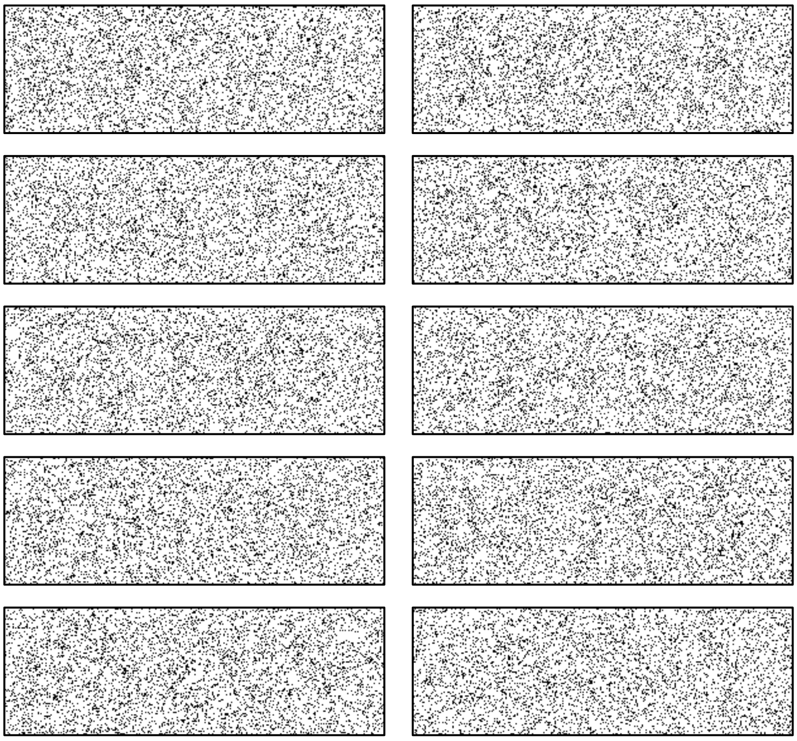
Similar to the previous experiment, the main objective in this scenario is to identify the impact of the total number of frames on the output results in terms of recognition success rates. To achieve this goal, the same experiment will be repeated, but this time the total number of input frames will be doubled to 10 frames, using the same OSR of

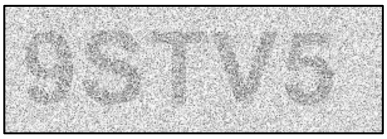
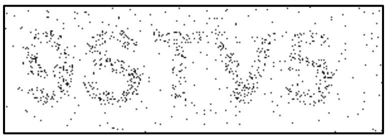
25% and the same BNR of 15%. Like the previous scenario, in this experiment 256 levels of greyscale need to be divided into 10 various levels because this time there are only 10 frames, as it has been shown in Figure 4.24.

Figure 4.24: A representation of the 256 binary bits divided into 10 shades of greyscale.

In this series of experiments, our aim is to analyse the impact that the number of frames has on the final output result, usefully comparing a 5-frame scenario with a 10-frame scenario and judging which superimposed output result has the higher clarity. Table 4.6 shows 10 different consecutive VICAP generated images with an OSR of 25% and a BNR of 15%.

Table 4.6: The generation of 10 CAPTCHA frames with an OSR of 25% and a BNR of 15%, followed by the superimposed output images in 8-bit/pixel greyscale format and 1-bit/pixel black and white image.

VICAP Frames (10 images)	

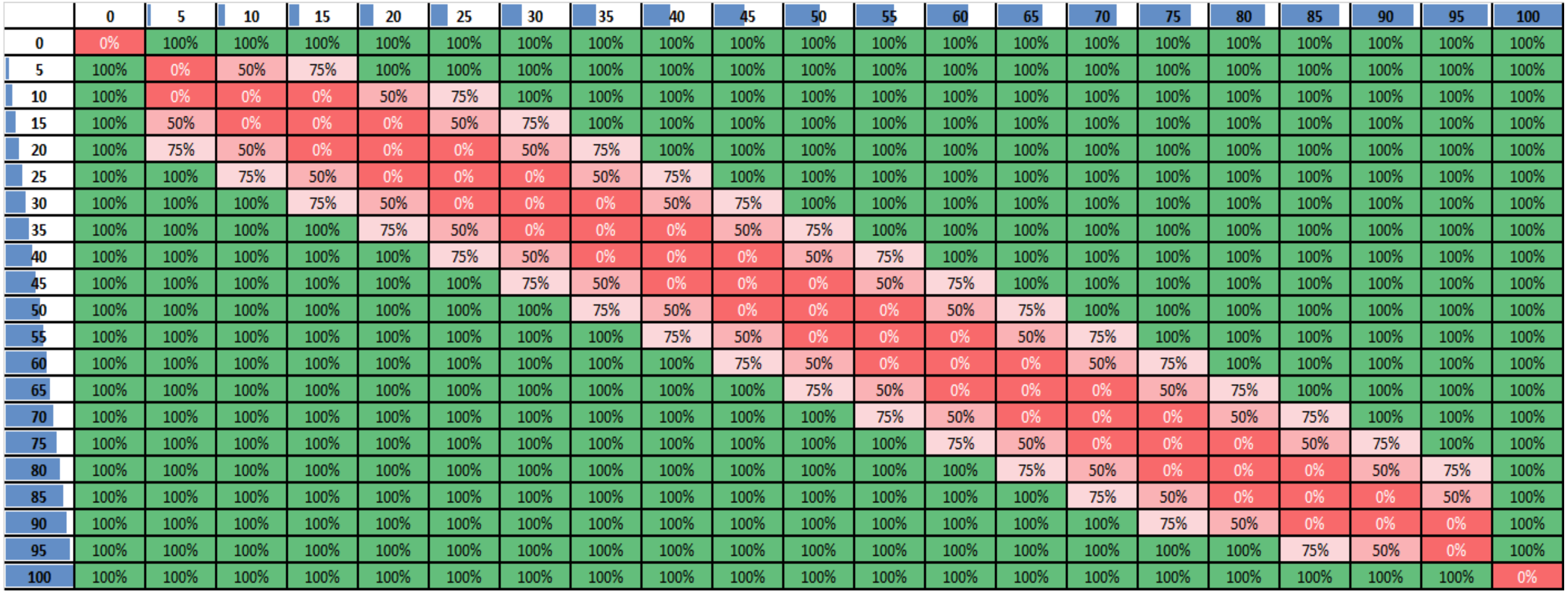
Output Superimposed results 8-bit/Pixel greyscale & 1-bit/Pixel Black and White	 

In this second experiment, each individual frame or CAPTCHA image does not show any useful information at first glance, and if every individual frame was analysed, no meaningful information would be retrieved. This is because, in every single frame, only a portion of the object pixels has been presented. In order to retrieve meaningful information from these frames, they must be played out consecutively before the human eye. As it can be seen from the output results in Table 4.6, the string '9STV5' is seen with a higher level of clarity compared to the last experiment. Consequently, the HRSR in this experiment would be around 75%, whilst the CRSR would be around 50%. More information about this comparison will be provided in the next section. From this experiment, it can be concluded that, by increasing the total number of frames, a clearer image will be produced, which can then be picked up by a human user's visual system with a high rate of accuracy.

4.5.3 The Impact of the Variable OSR vs. Changes in the Background Noise Level on CRSR

From the above experiments, there is a critical issue that must be addressed, which is the relationship between the OSR and the BNR. The key point to emphasise is the importance of choosing an appropriate density of object pixels compared to background noise. By doing this, we can enable the HVS to distinguish between object pixels and background noise. It is therefore vital to determine an appropriate ratio of OSR/BNR to ensure usability, but at the same time not compromise on security to ensure the new CAPTCHA is robust against computer attacks. Figure 4.14 presents 441 experimental results measuring the impact that different combinations of OSRs and BNRs have in relation to computer character recognition rates on VICAP images. Every value presented in this table is based on the average value for 10 randomly selected CAPTCHA images. After being superimposed and rendered using the CTA, these have been passed on to the

OCR programme. In total $[10 \times 441 = 4410]$ CAPTCHA experiments have been conducted and the results are presented as their average value in Figure 4.26. This research includes over 4,000 simulation experiments, which were conducted using a variety of OSRs and BNRs. The final images of these were then tested by computer recognition software. To make the research thorough and wide-ranging, the OSR was selected at a rate of 0%, to begin with, and was then increased by 5% granularly until it reached 100%. Similarly, the BNR started at 0% and was then increased by 5% each time until it reached 100%.



	0	5	10	15	20	25	30	35	40	45	50	55	60	65	70	75	80	85	90	95	100
0	0%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%
5	100%	0%	50%	75%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%
10	100%	0%	0%	0%	50%	75%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%
15	100%	50%	0%	0%	0%	50%	75%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%
20	100%	75%	50%	0%	0%	0%	50%	75%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%
25	100%	100%	75%	50%	0%	0%	0%	50%	75%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%
30	100%	100%	100%	75%	50%	0%	0%	0%	50%	75%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%
35	100%	100%	100%	100%	75%	50%	0%	0%	0%	50%	75%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%
40	100%	100%	100%	100%	100%	75%	50%	0%	0%	0%	50%	75%	100%	100%	100%	100%	100%	100%	100%	100%	100%
45	100%	100%	100%	100%	100%	100%	75%	50%	0%	0%	0%	50%	75%	100%	100%	100%	100%	100%	100%	100%	100%
50	100%	100%	100%	100%	100%	100%	100%	75%	50%	0%	0%	0%	50%	75%	100%	100%	100%	100%	100%	100%	100%
55	100%	100%	100%	100%	100%	100%	100%	100%	75%	50%	0%	0%	0%	50%	75%	100%	100%	100%	100%	100%	100%
60	100%	100%	100%	100%	100%	100%	100%	100%	100%	75%	50%	0%	0%	0%	50%	75%	100%	100%	100%	100%	100%
65	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	75%	50%	0%	0%	0%	50%	75%	100%	100%	100%	100%
70	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	75%	50%	0%	0%	0%	50%	75%	100%	100%	100%
75	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	75%	50%	0%	0%	0%	50%	75%	100%	100%
80	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	75%	50%	0%	0%	0%	50%	75%	100%
85	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	75%	50%	0%	0%	0%	50%	100%
90	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	75%	50%	0%	0%	0%	100%
95	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	75%	50%	0%	100%
100	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	0%

Figure 4.25: Comparison of different object sampling rates-OSR (horizontal rows) vs. backgrounds noise ratio-BNR (vertical columns) and the output results in terms of computer recognition success rate (in percentage).

4.6 Integrating VICAP into Web pages

The idea behind designing VICAP was to improve the security of websites and online forms. In order to achieve this, we have designed VICAP based on user-friendliness, and it can be integrated into almost all types of online forms and web pages. The VICAP programme itself was designed using a C# programming language and the actual programme has been placed on our CAPTCHA data web server. However, by using a simple scripting code, the VICAP protocol can be integrated into any third-party webpage. The VICAP data server can be connected to any third-party web server using JavaScript or PHP programming languages. Figure 4.26 shows the process of communication between the VICAP data server and a third-party web server using a secure communication channel.

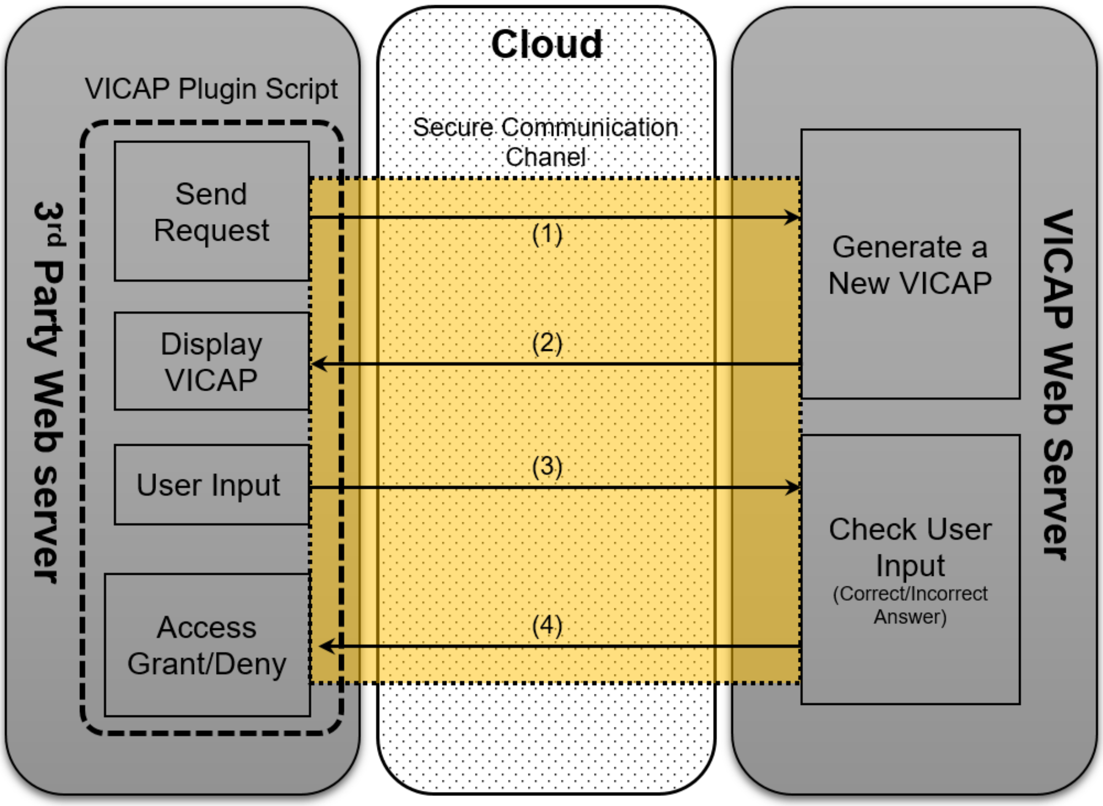
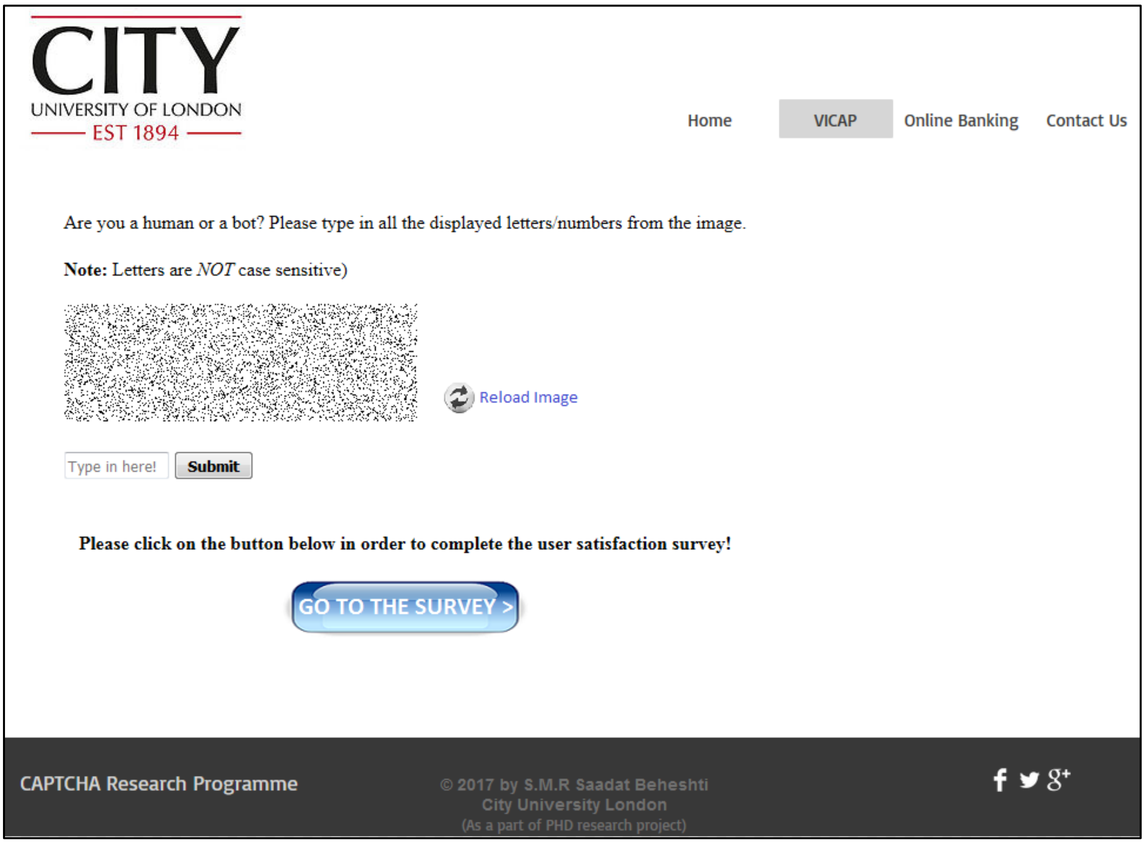


Figure 4.26: The process of integrating VICAP into a third-party web server. The requested information will be sent to the CAPTCHA server using secure communication channel. The CAPTCHA server will authenticate the response and grand or decline the access using same communication channel.

As it can be observed from Figure 4.26, VICAP plugin can be imported into a third-party web server using a simple script called ‘VICAP Script.’ VICAP script is a short piece of code that will pass on the VICAP server using a secure communication channel. This scripting code can be placed anywhere within the online form of the third-party web server. Thus, when a user wishes to complete the online form and comes to the authentication test, the CAPTCHA test can be called from the CAPTCHA server via the scripting code. Figure 4.27 demonstrates how the VICAP challenge will be integrated into an online web form.



Are you a human or a bot? Please type in all the displayed letters/numbers from the image.

Note: Letters are *NOT* case sensitive)

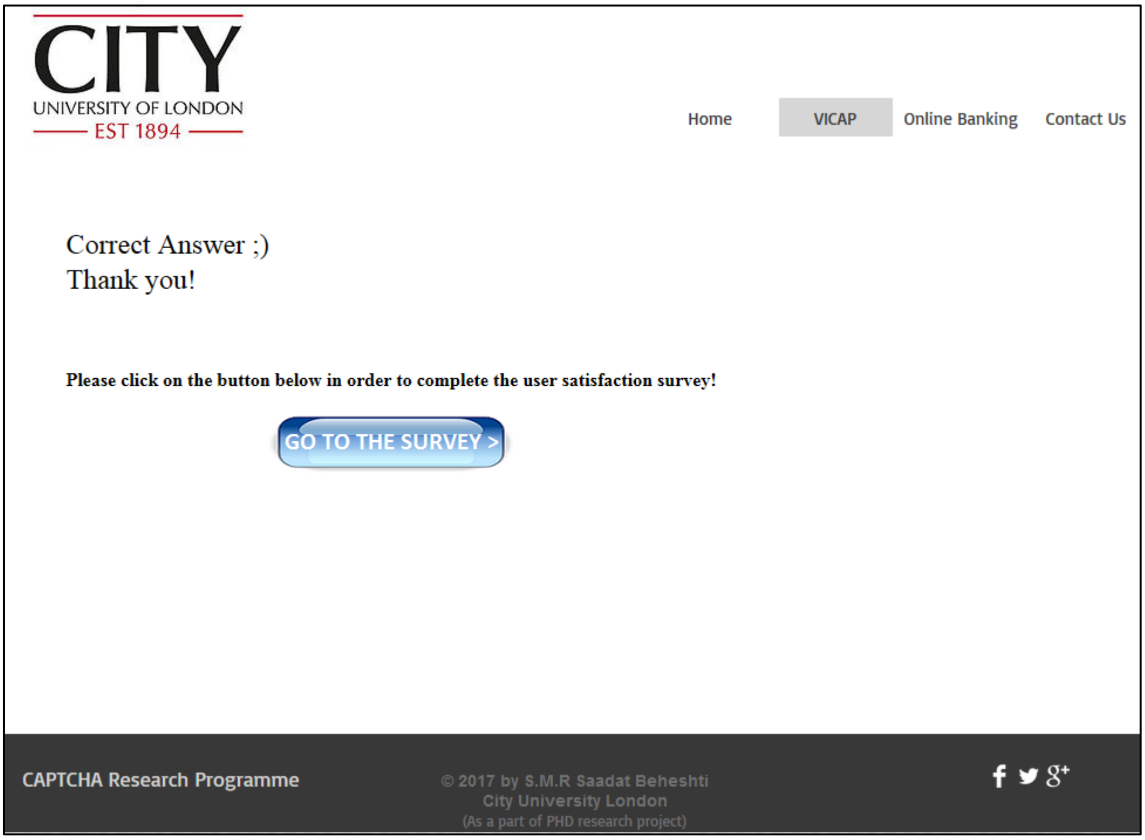
Please click on the button below in order to complete the user satisfaction survey!

CAPTCHA Research Programme

© 2017 by S.M.R Saadat Beheshti
City University London
(As a part of PHD research project)

Figure 4.27: Screenshot of the VICAP challenge integrated into an online application form.

The process will start by sending a ‘Request Signal’ to the CAPTCHA server, which will then generate a new CAPTCHA challenge and send it to the client web server. The user will then be able to see the CAPTCHA challenge and he or she will be required to type out the correct answer. The input response from the user will then be sent back to the CAPTCHA server, where it will be identified as correct or incorrect. If the user’s response is correct, the CAPTCHA server will send a ‘True Flag’ back to the third-party client’s web server, and access will be granted to the user. If the user’s answer is incorrect, the CAPTCHA server will send a ‘False Flag’ back to the client-server, which will then deny the user access or ask them to try again. Figure 4.28 demonstrates a correct response from the user to the VICAP test. This CAPTCHA has been accepted and access has been granted.



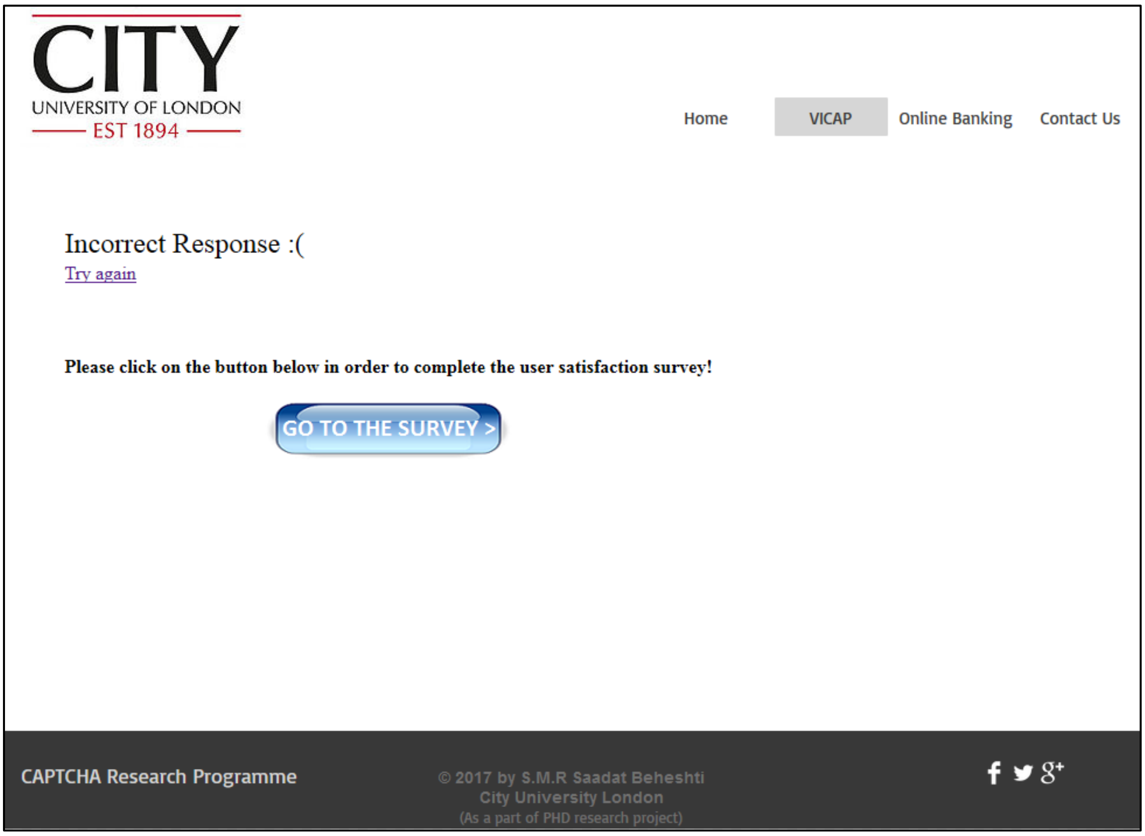
Correct Answer ;)
Thank you!

Please click on the button below in order to complete the user satisfaction survey!

© 2017 by S.M.R Saadat Beheshti
City University London
(As a part of PHD research project)

Figure 4.28: User has given the correct answer to the CAPTCHA challenge and the CAPTCHA has been accepted.

However, as it is shown in Figure 4.29, if the user’s response is incorrect, the system will show an ‘Incorrect Response’ message to the user. In this case, the user will be offered another attempt at the CAPTCHA challenge. Since the IP address of the user is recorded every time he or she tries to connect to the CAPTCHA server, it would be possible for the system to generate new sets of characters for every new try.



Incorrect Response :(

Try again

Please click on the button below in order to complete the user satisfaction survey!

GO TO THE SURVEY >
CAPTCHA Research Programme
© 2017 by S.M.R Saadat Beheshti
City University London
(As a part of PHD research project)

Figure 4.29: An incorrect response has been provided by the user and the CAPTCHA challenge is rejected, denying the user access.

4.7 Problem Associated with the VICAP v.1

As has been outlined, IM is known to cause the phenomenon POV. Forming a final object based on the presentation of a series of frames depends on the difference in the density of the object pixels and the background pixels. The way our visual system can distinguish between these two sets of pixels is by analyzing the density of all presented pixels in a single frame basis, then by capturing the initial frame in the IM and comparing it with the subsequent frame. As it can be understood from Equation (1) and (2) in section 3.5.2, for the human eye to distinguish between the object pixels and the background noise pixels, the entire process of displaying the CAPTCHA frames needs to be repeated. This process of repeating will cause our visual system to distinguish between the pixel density of the object and the background noise. Subsequently, based on the density information of the pixels, the human eye would be able to recognise the final object. However, the main question raised here is: what if a computer programme also compares the individual pixel frequencies in terms of multiple frames? By mapping these frequency values to the location of the pixels, in practice, the computer programme would be able to reconstruct the object too.

To address this problem, the proposed CAPTCHA model has been tested using the CTA. This application works in a very similar way to the HVS. It calculates the pixel occurrence frequencies according to the weight of each pixel, and by mapping the output results against different shades of grey, would be able to simulate the final image of the object in greyscale (8-bit/pixel) as well as in black and white (1-bit/pixel). This test will be done by making a graphical representation of the most frequent pixels by analyzing the pixel value based on a single frame and then expanding the calculation to the rest of the frames. In this experiment, we have selected 10 consecutive frames. Thus, as it can be observed from Figure 4.30, the developed application is easily able to retrieve the hidden information from 10 different VICAP frames and can recover the final object image very clearly.

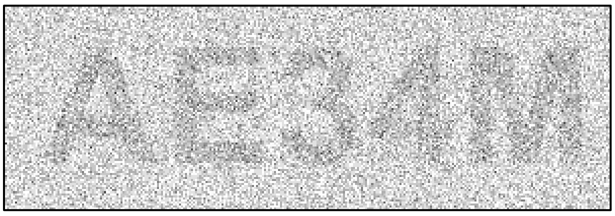


Figure 4.30: Final integrated output result from superimposing 10 CAPTCHA frames shown as an 8-bit/pixel greyscale image. A string of 'AE34M' can be clearly retrieved by OCR recognition software.

After superimposing a number of the original frames, it would be easy to render the final graphical representation of pixel densities according to the location of the pixels. As discussed above, an 8-bit/pixel greyscale image would be rendered using the CTA. After that, by applying some basic image-processing filters, we would be able to reform the characters, which would then be easily recognisable to an OCR programme.

As Figure 4.31 shows, various stages are involved in breaking and hacking VICAP v.1 as follows. Stage-1 involves superimposing a number of VICAP frames using the CTA to render the CAPTCHA image. Stage-2 would then convert the greyscale image into a binary image using threshold values. Stage-3 would then apply image-processing filters to remove the background noise from the image. Stage-4 would then utilise 'dot removal' and 'segmentation' algorithms to remove unwanted black pixels from the background and to separate individual characters. This also converts white pixels on the object to black to make the object more visible and more understandable for OCR software. Finally, Stage-5 recognises the characters produced using OCR software to decipher the final answer, as shown below.

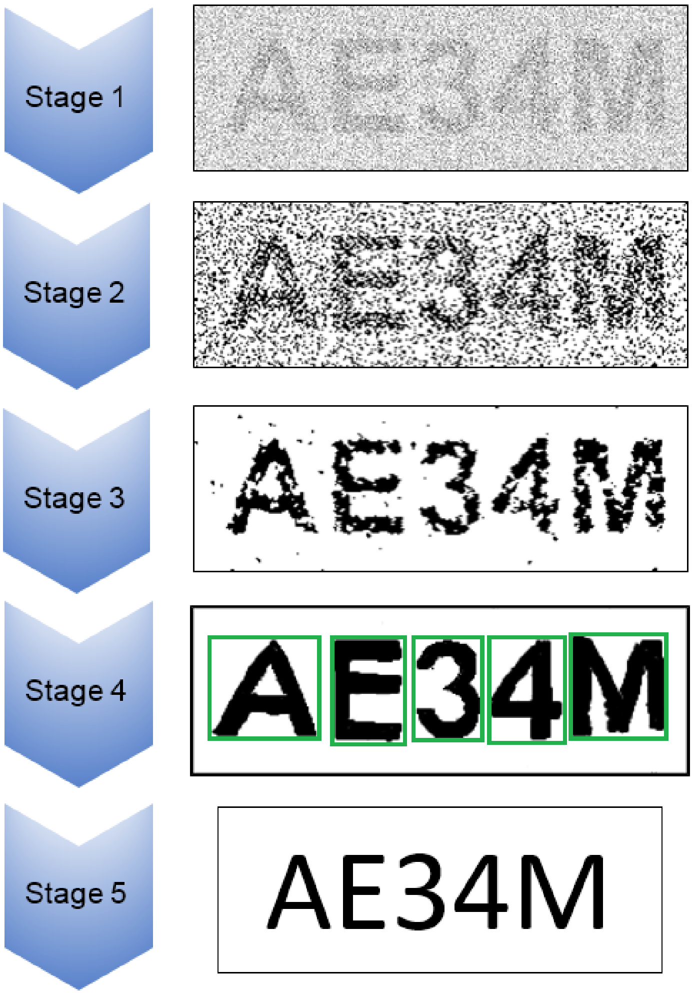


Figure 4.31: Various stages involved in breaking VICAP v.1. Stage-1: Superimposition. Stage-2: Binarisation. Stage-3: Background noise removal. Stage-4: Dot-removal and Segmentation. Stage-5: Using OCR and deciphering the correct answer.

This security hole is a significant area of weakness in the first generation VICAP model. Therefore, to rectify this issue, our second CAPTCHA design (labelled version 2.0) was developed. This will be discussed in the next chapter.

4.8 Conclusions

In this chapter, a novel CAPTCHA model was introduced called Visual Integration CAPTCHA (VICAP). As discussed previously, the main idea of VICAP is to utilise the ability of human users to superimpose and integrate a reel of fleeted frames into a uniform pattern, which will then represent a final image. This unique ability belongs to humans only, and no current recognition techniques are believed to be able to imitate this skill. In order to create this superimposed image, a state-of-the-art application was introduced called the CAPTCHA-Generator Application, which can render and generate a number of frames according to pre-determined conditions. Binarisation, using an OSR, and adding background noise are some of those conditions that will create POV in the brain. In order to test our new VICAP model, a state-of-the-art application was developed called the CAPTCHA-Test Application. As part of the VICAP evaluation programme, a comprehensive set of experiments was conducted under laboratory conditions. The output results from 5 and also 10 superimposed frames have been presented. We have also conducted a series of experiments to discover the impact that altering the OSR and the BNR has on the CRSR. As the output results confirm, having a large OSR to a small ratio of background noise (or vice versa) will increase the chances of CRSRs being high. This chapter has ended by discussing the possible lack of security of our proposed VICAP model, which has proved to be a significant weakness to the model. The next chapter will outline a new parameter called Original-to-Random Output Data (ORO), the purpose of which is to improve the lack of security of VICAP v.1.

Chapter 5

Original-to-Random Frames CAPTCHA Model (VICAP v.2)

5.1 Introducing Original-to-Random Output Data (ORO) Parameter

In our second proposed CAPTCHA model, the process of generating CAPTCHA frames is very similar to the first model. The only difference is that in the second model, there will be two production lines working concurrently. In what will be called Production Line-1, CAPTCHA images, which are original frames containing object information using the symbol 'O', are generated with the same procedure as before, while at the same time in what will be called Production Line-2, random frames, shown using the symbol 'R,' are generated, which contain random pixels at a rate of n. N is the same value as the BNR in the CAPTCHA frames of Production Line-1. Since both pixel frames, namely the background from Production Line-1 and Production Line-2, are generated from the same random process and thus have the same characteristics, they become indistinguishable. This increases the robustness of the model. There is currently no available recognition software that would be able to distinguish between these two sequences. An example of these two series of frame generator engines is given below:

Production line-1: $Original\,frames = \{O_1, O_2, O_3, \dots, O_m\}$

Production line-2: $Random\,frames = \{R_1, R_2, R_3, \dots, R_m\}$

Here, a key parameter is introduced called Original-to-Random Output (ORO) frames, which play a key role in the second CAPTCHA design. The rate of ORO frames defines the properties of the mixing procedure by specifying the percentage of random and original frames in the final sequence. A higher percentage of ORO translates to a larger number of original frames and a smaller number of random frames in the sequence and vice versa. An example of how ORO can be used to mix these two series of frames is shown below:

$$Output\ frame\ sequence\ = \{R_1, R_2, R_3, \boldsymbol{O_1}, R_4, \boldsymbol{O_2}, R_5, \dots\}$$

As it can be observed above, the original frames are being mixed with random frames with $ORO \simeq 30\%$. This mixing procedure is performed on a random basis. Therefore, in practice, it would not be possible to detect the ordering of the frames as a means to separate original frames from random ones. However, the superior properties of the HVS can superimpose the structured information in the presented frames, and thus distinguish the object from the background noise. Figure 5.1 shows an example of 10 output frames generated and rendered using an OSR of 25%, a BNR of 15%, and an ORO parameter set to 20%. As it can be seen from Figure 5.1, by analysing every single frame, no useful information can be observed. Yet, by running all frames at high speed, it is possible for the HVS to recognise the hidden object in the frame sequence.

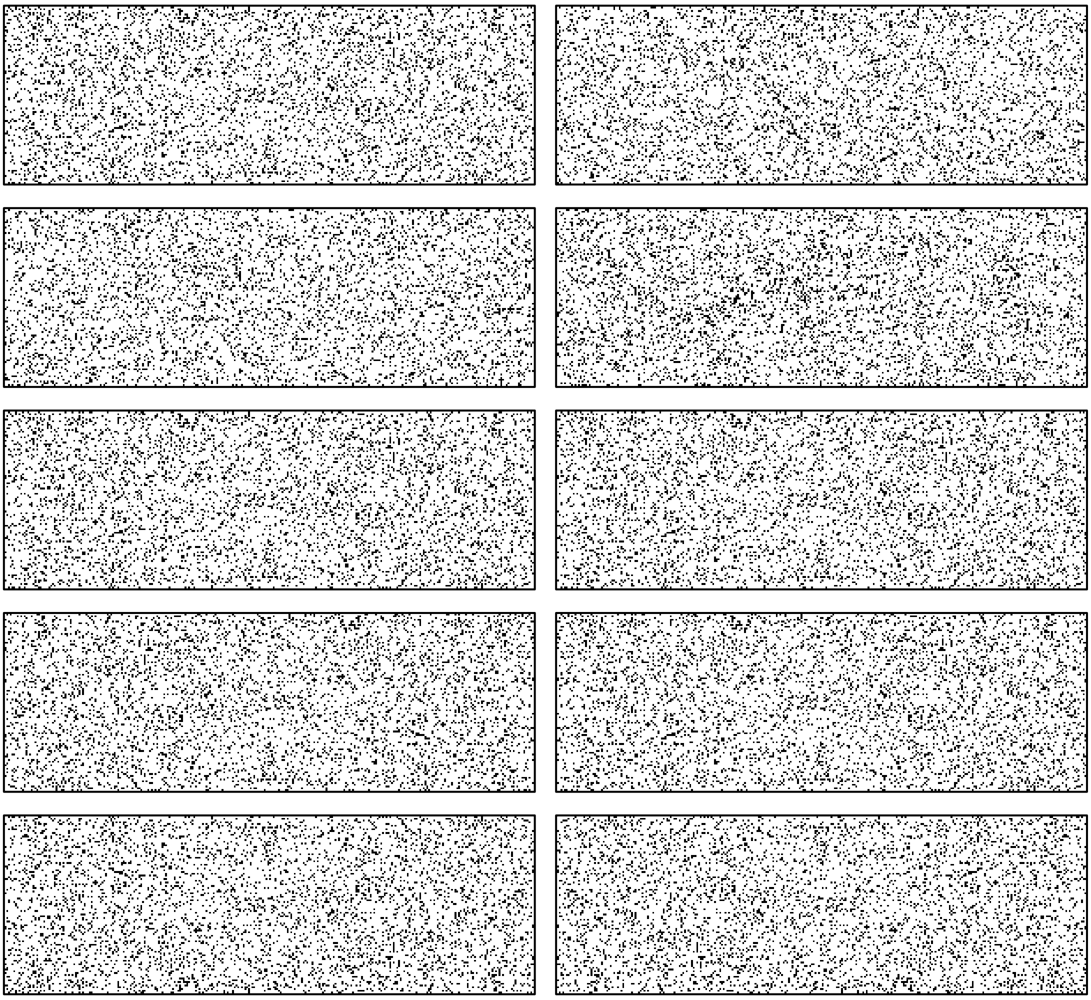


Figure 5.1: An example of 10 individual CAPTCHA frames generated and rendered by the VICAP-Generator Application with an OSR of 25%, a BNR of 15%, and an ORO of 20%.

As discussed previously, having different OSRs and BNRs affects the level of robustness of the proposed CAPTCHA model. Thus, as it was shown in the previous chapter in Figure 4.25, an OSR of 25% and a BNR of 15% from the VICAP v.1 gives a CRSR of 50%. When introducing VICAP v.2, the CRSR drops rapidly to near 0%, as shown in Figure 5.2 and Figure 5.3. The final simulation output results for the CRSR for VICAP v.2 is shown in these Figures using the CTA for an 8-bit/pixel greyscale format and a 1-bit/pixel black and white format. From these, we can see that no useful information can be retrieved from the series of frames and thus, it is expected that the CRSR is almost close to 0%.

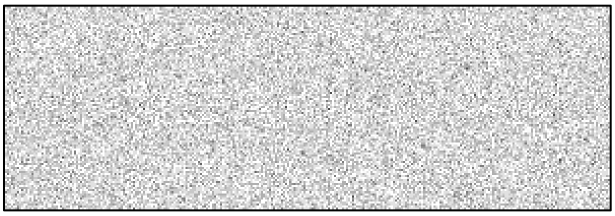


Figure 5.2: The final integrated output result from superimposing 10 VICAP frames, shown as an 8-bit/pixel greyscale image. No useful information can be retrieved in the second scenario.

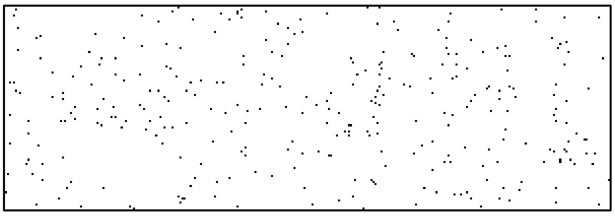


Figure 5.3: The final integrated output result from superimposing 10 VICAP frames, shown as a 1-bit/pixel black and white image. No useful information can be retrieved in the second scenario.

5.2 VICAP v.2 Attacks and Security Analysis

In this section, our aim is to discuss and analyse the most important threats that are currently affecting existing CAPTCHA models, and identify whether or not these kinds of attacks will have any impact on our proposed CAPTCHA model. As previously explained, text-based CAPTCHAs are the most vulnerable type of CAPTCHAs to character recognition attacks. We have explained various aspects of the OCR and the Human Character Recognition (HCR), methods that are currently the most threatening to these types of CAPTCHAs.

5.2.1 Optical Character Recognition Techniques and Vertical Segmentation

OCR software is an automated programme that can decipher a variety of glyphs and images, and then convert them into text. OCR processes start by scanning an entire image and then separating all the glyphs, a process known as ‘vertical segmentation.’ Vertical segmentation is a key element in any OCR software, and if the software cannot tell the glyphs apart or if they are overlapping, it would be very difficult for OCR programmes to distinguish them from each other. However, the most advanced OCR software uses incredibly sophisticated image-processing and shape recognition techniques that would enable them to recognise the boundaries of the glyphs and separate the glyphs. Yet, our proposed CAPTCHA model is based on the principle of POV and the properties of TI. Therefore, if an OCR programme were to analyse every individual frame, no useful or meaningful information could be retrieved. Thus, on an individual frame basis, the proposed CAPTCHA model is robust against any current character recognition applications at a rate of 100%. This robustness can be achieved by two main factors. Firstly, the characters and numbers are completely obscured by background noise using sampling techniques. More information about this technique was provided in section 4.3.2. Consequently, every single frame is made up of a number of random dots (or black pixels), which on their own contain no useful meaning. A single VICAP frame is shown in Figure 5.4.

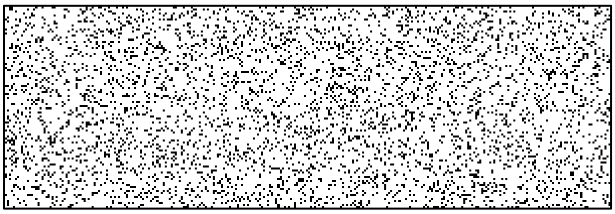
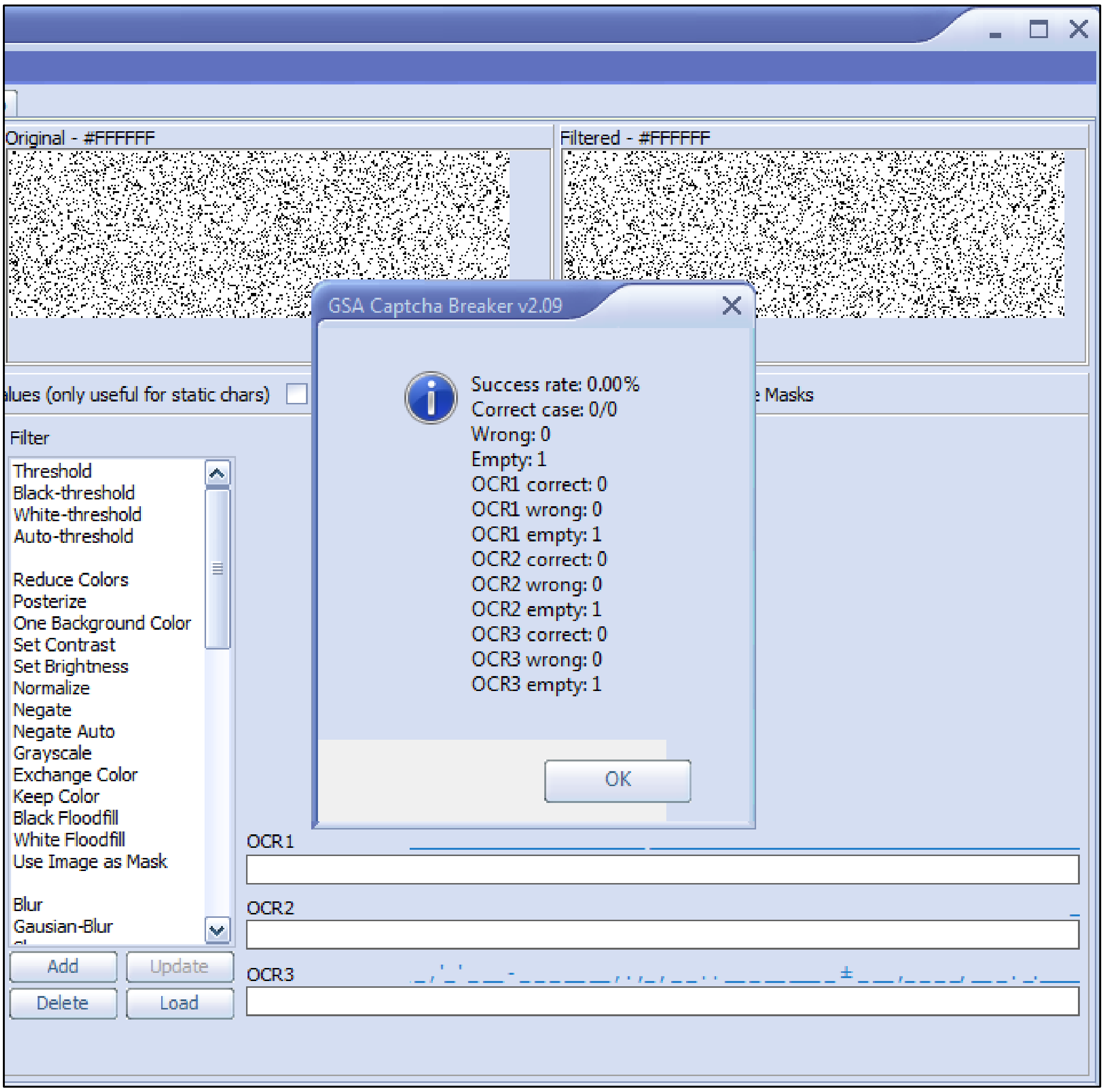


Figure 5.4: VICAP single frame containing random pixels.

Secondly, as was explained before, the way OCR programmes can decipher any text is by segmenting the characters and then applying pattern-matching techniques in order to recognise the text. Since all letters and numbers in our model are sampled and only a portion of their pixels are presented in every single frame, in practice it would not be possible for OCR software to perceive any meaningful information. Hence, no current OCR software would be able to segment the VICAP frames. Figure 5.5 presents a screenshot of one of the most advanced CAPTCHA decoders called GSA CAPTCHA Breaker, with three different OCR recognition engines working concurrently. As the output results confirm, the OCR software is showing a 0% recognition success rate for individual CAPTCHA frames. This type of CAPTCHA decoder will be explained in depth in section 5.3.



```
Success rate: 0.00%
Correct case: 0/0
Wrong: 0
Empty: 1
OCR1 correct: 0
OCR1 wrong: 0
OCR1 empty: 1
OCR2 correct: 0
OCR2 wrong: 0
OCR2 empty: 1
OCR3 correct: 0
OCR3 wrong: 0
OCR3 empty: 1
```

Figure 5.5: A screenshot of the GSA-CAPTCHA Breaker showing a 0% recognition success rate for individual frames.

5.2.2 Third Party Human Attacks

As has already been discussed, human attacks are even more threatening than computer attacks in the current day. Since the concept of having a CAPTCHA is based on the idea that only humans should be able to solve the test, human attackers and hackers pose a very significant threat to CAPTCHA security. As has already been outlined, a CAPTCHA image will be sent to a group of third-party human attackers using Instant Messaging Service (IMS). Human attackers will then see the CAPTCHA, solve it, and then send it back to the server using the same communication channel. This process of seizing CAPTCHA images and sending them to human attackers across the world, who then fill in the correct answers, is a very quick procedure. Indeed, in practice, this method is known to be the most accurate and cheapest type of CAPTCHA attack. Since our developed VICAP model is based on POV and psychophysics, the user is required to physically sit in front of their computer and stare at a sequence of images in order to create a final image in their visual system. For this reason, even capturing a screenshot and sending individual frames to human attackers will yield no useful information because nothing will be perceived apart from random noise.

5.2.3 Dictionary-Based Attacks

Dictionary-based attacks are another type of CAPTCHA attack that OCR engines use in order to find unknown words from a dictionary database. The vast majority of text-based CAPTCHAs are made up of a single word selected from a variety of resources, and then after being applied with a level of distortion, it is then presented as a CAPTCHA. Using a meaningful word in a CAPTCHA can be very risky. As we have shown in this research, it would be very easy to reconstruct any distorted characters and then run them against a dictionary application. Since our developed CAPTCHA uses only random characters and numbers, it is not dictionary-based. Thus, in practice, it would be impossible for computer programmes to find meaningful words by mapping output results to a dictionary database.

5.2.4 Text-Recognition Algorithm Working on Visual Traces

Another important security concern with most text-based CAPTCHAs is dictionary-based (or guessing) attacks. These types of CAPTCHA attacks recognise partial information from a string of characters, and based on the preceding and succeeding information, would be able to predict any missing elements. Then, by mapping and comparing the deciphered information against a dictionary database, the programme would be able to decipher the unrecognized word. In our proposed CAPTCHA model, the order of the frames is random and the way each individual frame is rendered and presented to the user is also random. Therefore, it is impossible for any text recognition programme to predict the future location of the pixels corresponding to the original character. If a CAPTCHA sequence follows a pattern, it would be possible for a bot to track changes and predict the final object based on that information. For instance, by looking at the example in Figure 5.6, we can visually appreciate that the symbol '?' can be declared as letter 'C' based on the preceding and succeeding information.

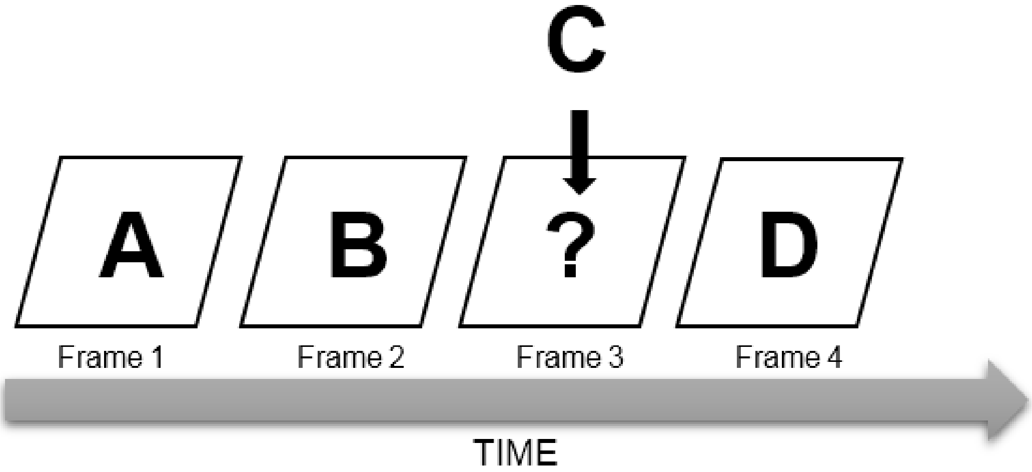


Figure 5.6: Uniform presentation of pixels. By presenting all the frames in the uniform sequence it will be easy to predict the missing frame.

However, in the second example in Figure 5.7, it is impossible to guess the value of symbol '?' This is because the presence of object pixels and background noise pixels in all the preceding and succeeding frames are selected randomly in the context of a single frame. The lack of uniform or regular patterns in the frame information prevents the use of predictive algorithms to guess the future position of pixels or decipher the hidden character.

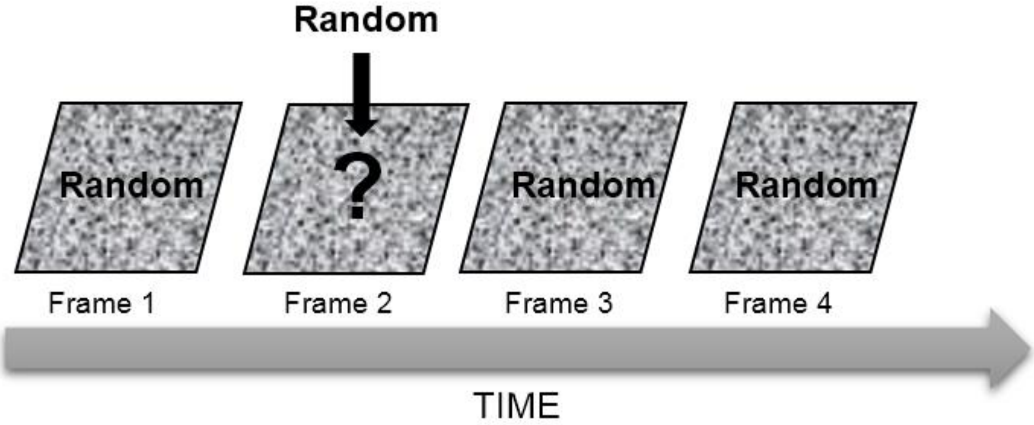


Figure 5.7: By presenting the pixels in a random sequence, it would not be possible for the tracing machine to predict the behaviour of the pixels by comparing the preceding and succeeding frames.

Since our proposed CAPTCHA model is not dictionary-based and every individual element of the CAPTCHA string is selected randomly, even if one or two elements are decoded, it would not be possible for computer recognition programmes to guess the entire CAPTCHA string.

5.3 VICAP v.2 Security and Robustness Experimental Results and Discussion

As discussed previously, the clarity of letters and numbers in the proposed CAPTCHA model directly depends on the value of the ORO parameter because this affects the recognition success rate for both human users and OCR software. In this section, the VICAP v.2 model was tested using different techniques under different conditions for both human users as well as OCR software in order to determine the optimal level for the ORO parameter. This would translate as the *threshold value* by which the CAPTCHA is easily solvable for human users, but at the same time, remains difficult (if not impossible) for computer attackers to solve. The approach to the first set of experiments is to play the proposed CAPTCHA model for the OCR program and by increasing the ORO value the recognition success rate will be measured. The process of increasing the ORO parameter will continue gradually until it reaches to 100% recognition success rate. In the second approach, the same experiment will be repeated but this time for human users. Similar to the first approach the value of ORO parameter starts from 0% and it will gradually increase until it reaches to 100%. Along with increasing ORO parameter, the human recognition success rate will be also measured and it will be noted as soon as it reaches the 100% success rate. The idea in these two series of experiments is to compare the complete recognition success rate threshold for both computers programs and human users.

5.3.1 Experiment 1: Computer Recognition Success Rate (CRSR)

To measure the level of robustness and security of the proposed CAPTCHA model, various experiments were conducted in laboratory conditions using a computer with Intel Core-i5 CPU and 3.20 GHz processor. 200 attacks were simulated using a state-of-the-art GSA-CAPTCHA Breaker (GCB) [63]. The aim of these experiments was to determine the resistance level (or threshold value) of the proposed VICAP model against different computer recognition attacks. The experiments began with ORO = 0% and the CRSR was measured in every single experiment. The ORO was increased by 10% until it reached 100%. To generate statistically meaningful results, for every single setting of the ORO parameter, a chunk of 20 randomly generated CAPTCHAs were fed into the GCB application.

Although the second version of the proposed VICAP model is based on the principle of POV and the properties of TI, after playing back and analysing the individual frame information, no useful information could be retrieved. Thus, on an individual frame basis, the model is 100% secure against any current character recognition application for two main reasons. Firstly, the characters and numbers are completely faded into background noise using sampling techniques. Consequently, every single frame is made up of random dots (or black pixels) which, on their own, have no meaning. Secondly, the way OCR programmes recognise text is by firstly segmenting the characters and then recognising them. Since the letters and numbers in our model are sampled, only a portion of their pixels appears in each frame, which does not carry sufficient information about the object. Hence, OCR software is not able to segment VICAP frames. To test the proposed CAPTCHA model, we needed to simulate a final output image, which implements the integration rules of Equations (1) and (2) from section 3.5.2. To get the final output results, we have used our CTA. As discussed previously, the ORO *threshold value* is the key parameter that is being tested and evaluated in this section. To better understand how the ORO parameter can affect the visibility and distinguishability of the characters and numbers in VICAP v.2, some of the superimposed output results are presented in Figure 5.8 to Figure 5.12. As it can be observed from the output results, the clarity of the characters is lowest when the ORO parameter is about 10%, and gradually increases as the value of the ORO parameter increases.

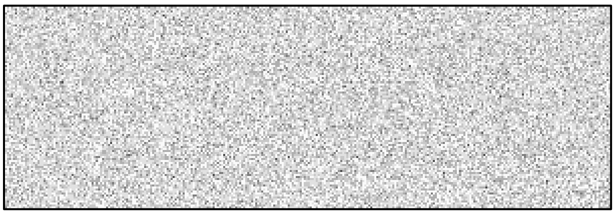


Figure 5.8: Superimposition rendered output image with ORO = 10%.

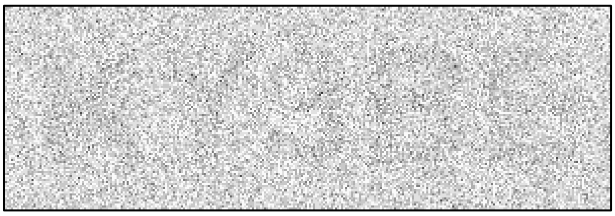


Figure 5.9: Superimposition rendered output image with ORO = 30%.

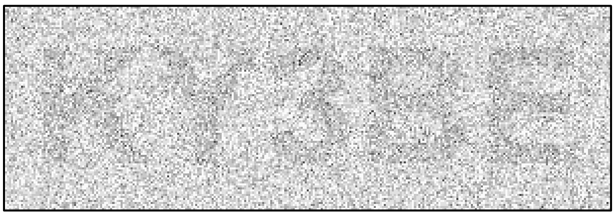


Figure 5.10: Superimposition rendered output image with ORO = 50%.

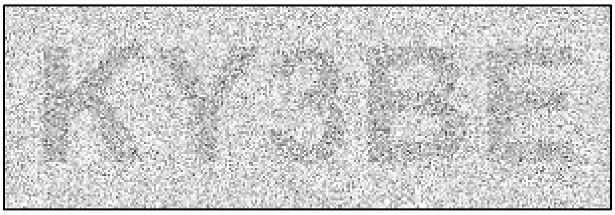


Figure 5.11: Superimposition rendered output image with ORO = 70%.

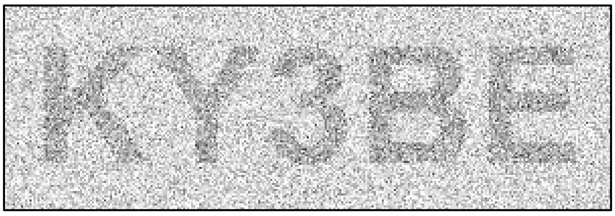


Figure 5.12: Superimposition rendered output image with ORO = 90%.

These experiments aimed to measure the security and robustness levels of the proposed CAPTCHA model v.2. Every single experiment consisted of superimposing images of 10 randomly selected VICAP frames with a specific ORO value as shown above. 10 different ratios of ORO parameters (from 0% to 100%) and 20 randomly rendered and superimposed VICAP images per ratio were tested. As stated before, individual VICAP frames were rendered and generated using a state-of-the-art CGA and after that, every group of frames were superimposed into a single image using the CTA.

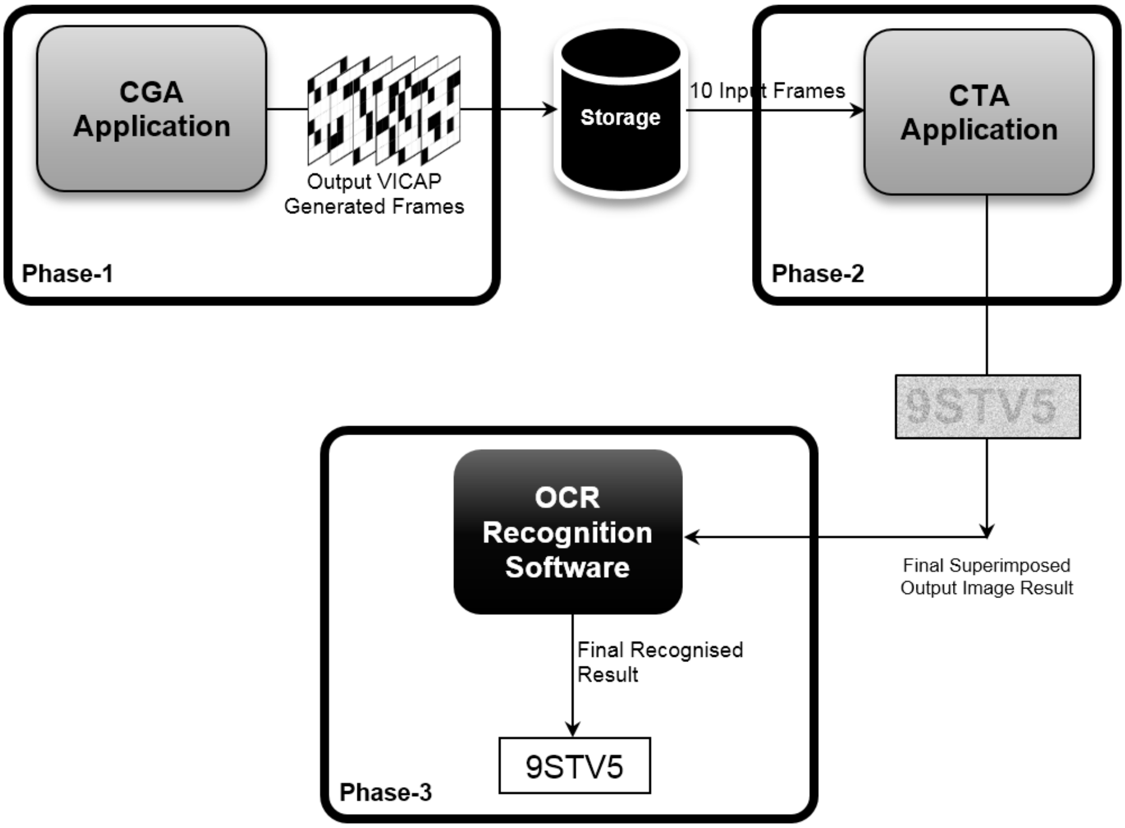
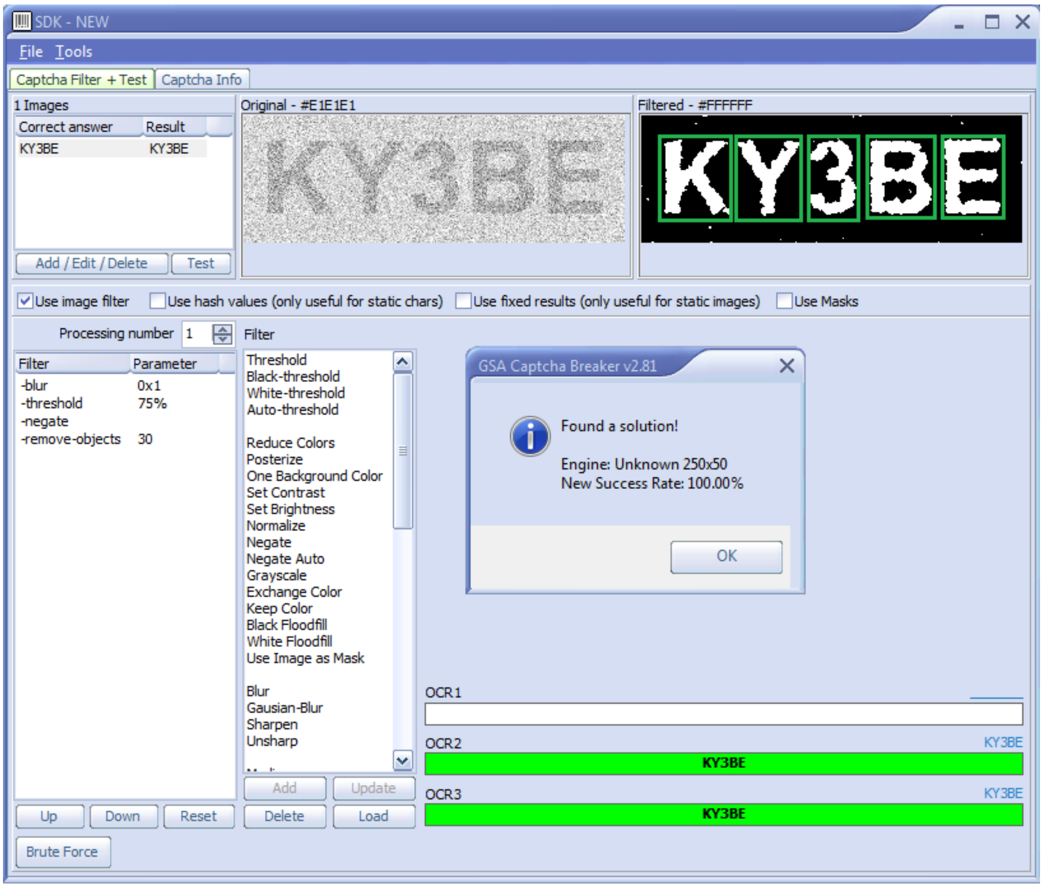


Figure 5.13: The figure represents three different phases involved in testing our proposed VICAP model. The CAPTCHA images will be produced in phase-1. Then they will superimpose and integrated using CTA application in phase-2. Finally, the superimposed output image will be used for recognition software in phase-3.

In order to conduct these experiments, VICAP frames first needed to process and superimposed to make them readable for OCR software. As mentioned earlier, it would not be possible to analyse every individual frame because it would give almost a 0% recognition success rate for the OCR software. The whole procedure to test our proposed CAPTCHA model will be done in three phases, as illustrated in Figure 5.13. Phase 1 is the CGA, which will generate all the CAPTCHA frames and store all of these in the storage folder. In order to analyse and superimpose the final output results, the CTA was used to feed 10 consecutive input frames into the pre-located storage using the CGA. After the final output superimposed image is rendered using the CTA, the output result is then ready to be fed into the OCR software to be recognised and deciphered. Figure 5.13 illustrates these three phases of this procedure.

After rendering and generating 200 different superimposed CAPTCHA images using the CTA, these were then passed onto the GCB to measure their security and robustness levels. As Figure 5.14 shows, the GCB could break our proposed model in about 35 seconds with a success rate of 100% when ORO = 90%. As the value of ORO parameter dropped, the probability of mixing original frames with random frames also decreased. As the number of original frames drops, less information about the object is presented. Therefore, it would be more difficult for OCR software to decipher useful information and subsequently, this would translate to a reduced recognition success rate for automated computer programmes. Figure 5.15 shows that computer recognition attacks are not able to break our proposed CAPTCHA model when the ORO parameter is equal to 20% or less.



1 Images

Correct answer	Result
KY3BE	KY3BE

Original - #E1E1E1

Filtered - #FFFFFF

☑ Use image filter ☐ Use hash values (only useful for static chars) ☐ Use fixed results (only useful for static images) ☐ Use Masks

Processing number 1

Filter	Parameter
-blur	0x1
-threshold	75%
-negate	
-remove-objects	30

GSA Captcha Breaker v2.81

Found a solution!

Engine: Unknown 250x50
New Success Rate: 100.00%

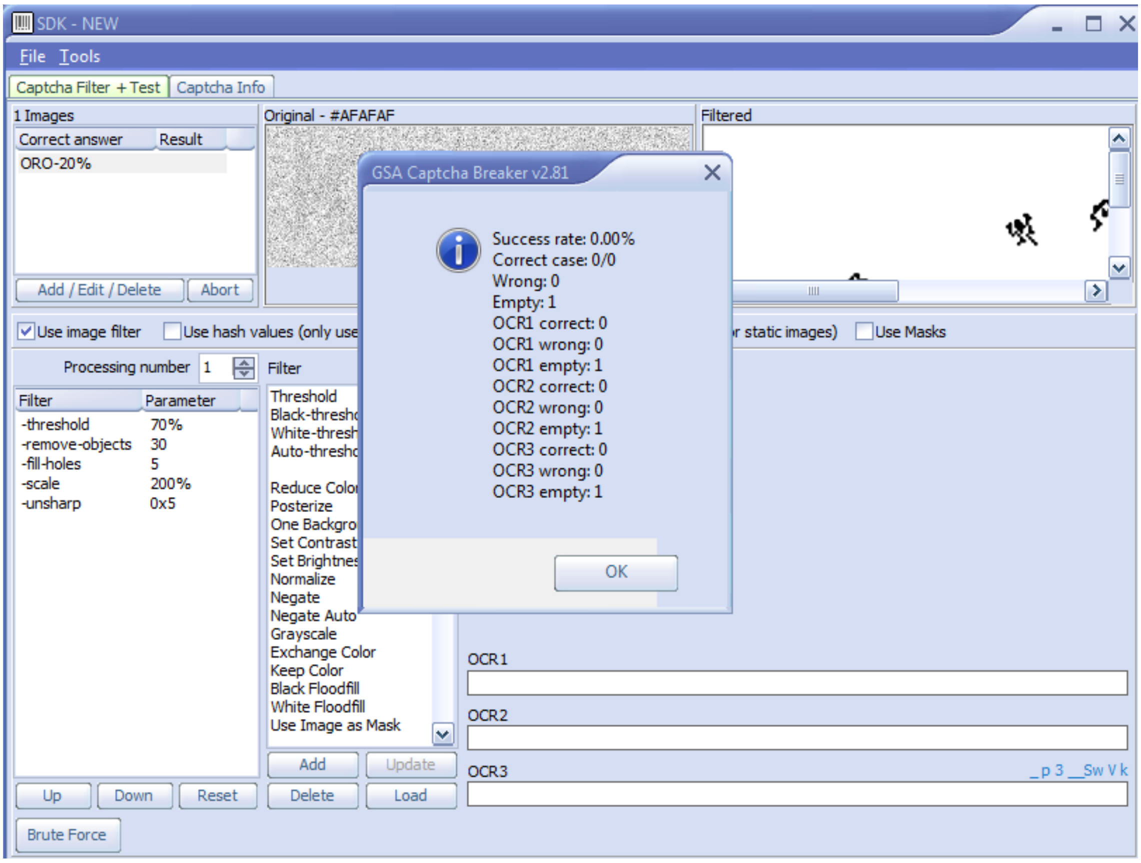
OK

OCR1

OCR2 KY3BE
KY3BE

OCR3 KY3BE
KY3BE

Figure 5.14: A screenshot of the GCB application. VICAP is recognised by the GCB application when the ORO = 90%, with a recognition success rate of 100%.



Success rate: 0.00%
Correct case: 0/0
Wrong: 0
Empty: 1
OCR1 correct: 0
OCR1 wrong: 0
OCR1 empty: 1
OCR2 correct: 0
OCR2 wrong: 0
OCR2 empty: 1
OCR3 correct: 0
OCR3 wrong: 0
OCR3 empty: 1

Filter	Parameter
-threshold	70%
-remove-objects	30
-fill-holes	5
-scale	200%
-unsharp	0x5

Figure 5.15: A screenshot of the GCB application. The GCB application is not able to retrieve any information from the object when the ORO = 20%. In this case, the recognition success rate is 0%.

Table 5.1 presents a comparison of different ORO values and the corresponding CRSRs. As the table shows, the value of ORO parameters can directly affect the CRSR. By increasing the ORO parameter, the CRSR would also increase. There would be a threshold value, which is determined to be ORO = 20% in this experiment, where the CRSR equals 0%. This threshold value plays a key role in this experiment because for values of ORO > 20%, computer recognition attacks would be able to break the proposed CAPTCHA model. Therefore, as we can see from Table 5.1, there is a significant jump in terms of CRSRs, going from 0% to 20% when the ORO increases from 20% to 30%.

Table 5.1: Comparison of CRSRs versus the ORO parameter.

ORO Parameters	Computer Recognition Success Rate
0%	0%
10%	0%
20%	0%
30%	20%
40%	50%
50%	80%
60%	90%
70%	100%
80%	100%
90%	100%
100%	100%

To find out whether the sudden increase in the CRSR when the ORO increases from 20% to 30% is instantaneous or gradual, we ran an extensive series of new experiments in the same laboratory conditions as explained before. The purpose of the first experiment was to evaluate the CRSRs in terms of whether or not it could decipher the entire CAPTCHA. We set the percentage increase of the ORO to 1% and created a test database of 20 randomly generated and superimposed VICAP images for each ORO value, resulting in a total of 220 images. As shown in Table 5.2, the CRSR stayed at 0% when the ORO equalled 29% or less, which then rose sharply to 20% when the ORO rate hit 30%.

Table 5.2: The change in CRSRs for an entire CAPTCHA string when varying the ORO parameters.

ORO Parameter	CRSR
20%	0%
21%	0%
22%	0%
23%	0%
24%	0%
25%	0%
26%	0%
27%	0%
28%	0%
29%	0%
30%	20%

Table 5.3: Change in the CRSRs for partial CAPTCHA character recognition when varying the ORO parameters.

ORO Parameters	CRSR
20%	0%
21%	0%
22%	0%
23%	0%
24%	0%
25%	0%
26%	0%
27%	1%
28%	3%
29%	7%
30%	28%

To explain the reason for this sudden jump in terms of the CRSRs, we turned our attention to analysing the individual character recognition rates in each of the CAPTCHA strings. The results of this are shown in Table 5.3. As we can see from Table 5.3, partial character recognition success rates are higher than CAPTCHA string recognition success rates for the same ORO levels. This is due to the fact that recognising more CAPTCHA characters increases the chances of recognising the entire CAPTCHA string. For example, as can be seen from Table 5.3, when the ORO = 30%, the partial CAPTCHA character recognition success rate is equal to 28%. While the entire CAPTCHA string recognition success rate is equal to 20%.

In order to elaborate on the methodology used to calculate partial recognition success rates, we present Table 5.4. As can be seen from this table, in the experiments where the value of the ORO parameter is equal to 30%, there are 4 occurrences where the CRSR equalled 100%, which successfully identified all five CAPTCHA characters - i.e. the entire string. Similarly, there is only one occurrence where four out of the five CAPTCHA characters are identified, thus producing a CRSR of 80%. Also, there is one case where three out of the five characters were identified, thus producing a CRSR of 60%. Finally, there was a trial with a CRSR of 20%, where only one out of the five CAPTCHA characters was identified. By calculating the weighted average values of all CRSRs at a specific ORO level (for example, 30%), the partial computer recognition success rate value was calculated (for example, 28%). At this ORO level, there are 4 occurrences where the CRSR equals 100%, where the entire CAPTCHA string was correctly recognised, corresponding to 20 recognised characters. There is one instance each for the CRSR equalling 80% (four correctly recognised characters), 60% (three correctly recognised characters), and 20% (one correctly recognised character). Thus, in a total of 20 trials (each containing five characters), 28 characters were correctly recognised (i.e. 28%).

Table 5.4: Partial recognition success rate calculation methodology.

No. of Experiments	ORO Parameters	CRSR	Average CRSR
1	26%	0%	0%
2	26%	0%	
3	26%	0%	
4	26%	0%	
5	26%	0%	
6	26%	0%	
7	26%	0%	
8	26%	0%	
9	26%	0%	
10	26%	0%	
11	26%	0%	
12	26%	0%	
13	26%	0%	
14	26%	0%	
15	26%	0%	
16	26%	0%	
17	26%	0%	
18	26%	0%	
19	26%	0%	
20	26%	0%	
1	27%	0%	1%
2	27%	0%	
3	27%	0%	
4	27%	0%	
5	27%	0%	
6	27%	0%	
7	27%	0%	
8	27%	0%	
9	27%	20%	
10	27%	0%	
11	27%	0%	
12	27%	0%	
13	27%	0%	
14	27%	0%	
15	27%	0%	
16	27%	0%	
17	27%	0%	

18	27%	0%	
19	27%	0%	
20	27%	0%	
1	28%	0%	3%
2	28%	0%	
3	28%	20%	
4	28%	0%	
5	28%	0%	
6	28%	0%	
7	28%	0%	
8	28%	0%	
9	28%	0%	
10	28%	0%	
11	28%	20%	
12	28%	0%	
13	28%	0%	
14	28%	0%	
15	28%	0%	
16	28%	0%	
17	28%	20%	
18	28%	0%	
19	28%	0%	
20	28%	0%	
1	29%	0%	7%
2	29%	0%	
3	29%	0%	
4	29%	0%	
5	29%	20%	
6	29%	0%	
7	29%	0%	
8	29%	40%	
9	29%	0%	
10	29%	0%	
11	29%	0%	
12	29%	0%	
13	29%	0%	
14	29%	0%	
15	29%	0%	
16	29%	20%	
17	29%	0%	

18	29%	20%	
19	29%	40%	
20	29%	0%	
1	30%	0%	28%
2	30%	100%	
3	30%	0%	
4	30%	100%	
5	30%	0%	
6	30%	0%	
7	30%	20%	
8	30%	0%	
9	30%	0%	
10	30%	80%	
11	30%	0%	
12	30%	0%	
13	30%	100%	
14	30%	0%	
15	30%	0%	
16	30%	100%	
17	30%	0%	
18	30%	0%	
19	30%	60%	
20	30%	0%	

Consequently, from Table 5.4, it is possible to calculate the number of successful character recognition trials for the entire CAPTCHA string (of all five characters). By looking into each ORO category and only picking the trials where *all* five characters are identified successfully and also calculating the total average value for those successful trails, it would be possible to calculate the complete recognition success rate. For instance, at an ORO rate of 30%, there were only four trials where all five characters were correctly recognised, thus producing a CRSR of 20% for the complete CAPTCHA string. Similarly, for an ORO rate of 29%, there were no instances where the entire CAPTCHA was recognised, with only partial recognition being achieved, resulting in a partial character recognition success rate of 7% and a complete CAPTCHA string recognition success rate of 0%.

As the value of the ORO increases so does the CRSR value. It is interesting to note that when ORO equals 26% or lower, CRSRs are at 0%. However, above 26% there is a non-linear relationship in the increase of the CRSR. This is exemplified by the massive jump in the ORO values of 29% and 30%, where the CRSR quadruples. When revisiting the results of Table 5.2 for the entire CAPTCHA string, it is logical that the CRSR is at 0% up to an ORO level of 29%, as only a total of 7% of the complete set of individual CAPTCHA characters were successfully recognised. As the individual CAPTCHA character CRSR rises to 28% at an ORO level of 30%, this leads to the CAPTCHA string CRSR rising to 20%. To validate the findings, the proposed CAPTCHA model was tested against other popular CAPTCHA decoders. As explained previously, GCB is software that uses three different OCR engines to recognise every single character. CAPTCHA Sniper (CS) is another type of CAPTCHA decoder that works very similarly to the GCB, and both are based on the same concept. By applying CS and the GCB, the performance of some of the most popular CAPTCHA decoders such as DeCaptcher, DeathByCaptcha, and Bypass Captcha were evaluated [64] [65]. Table 5.5 presents some of the output results from the experiments using the named CAPTCHA decoders.

Table 5.5: Recognition output results for different CAPTCHA decoders.

CAPTCHA Decoders	ORO	CRSR
GSA Captcha Breaker	0.0%	0.0%
Captcha Sniper	0.0%	0.0%
DeCaptcher	0.0%	0.0%
DeathByCaptcha	0.0%	0.0%
Bypass Captcha	0.0%	0.0%
GSA Captcha Breaker	10.0%	0.0%
Captcha Sniper	10.0%	0.0%
DeCaptcher	10.0%	0.0%
DeathByCaptcha	10.0%	0.0%
Bypass Captcha	10.0%	0.0%
GSA Captcha Breaker	20.0%	0.0%
Captcha Sniper	20.0%	0.0%
DeCaptcher	20.0%	0.0%

DeathByCaptcha	20.0%	0.0%
Bypass Captcha	20.0%	0.0%
GSA Captcha Breaker	30.0%	20.0%
Captcha Sniper	30.0%	25.0%
DeCaptcher	30.0%	20.0%
DeathByCaptcha	30.0%	15.0%
Bypass Captcha	30.0%	20.0%
GSA Captcha Breaker	40.0%	50.0%
Captcha Sniper	40.0%	55.0%
DeCaptcher	40.0%	55.0%
DeathByCaptcha	40.0%	45.0%
Bypass Captcha	40.0%	50.0%
GSA Captcha Breaker	50.0%	80.0%
Captcha Sniper	50.0%	75.0%
DeCaptcher	50.0%	85.0%
DeathByCaptcha	50.0%	70.0%
Bypass Captcha	50.0%	85.0%
GSA Captcha Breaker	60.0%	90.0%
Captcha Sniper	60.0%	95.0%
DeCaptcher	60.0%	90.0%
DeathByCaptcha	60.0%	80.0%
Bypass Captcha	60.0%	95.0%
GSA Captcha Breaker	70.0%	100.0%
Captcha Sniper	70.0%	100.0%
DeCaptcher	70.0%	95.0%
DeathByCaptcha	70.0%	90.0%
Bypass Captcha	70.0%	100.0%
GSA Captcha Breaker	80.0%	100.0%
Captcha Sniper	80.0%	100.0%
DeCaptcher	80.0%	100.0%
DeathByCaptcha	80.0%	95.0%
Bypass Captcha	80.0%	100.0%
GSA Captcha Breaker	90.0%	100.0%

Captcha Sniper	90.0%	100.0%
DeCaptcher	90.0%	100.0%
DeathByCaptcha	90.0%	100.0%
Bypass Captcha	90.0%	100.0%
GSA Captcha Breaker	100.0%	100.0%
Captcha Sniper	100.0%	100.0%
DeCaptcher	100.0%	100.0%
DeathByCaptcha	100.0%	100.0%
Bypass Captcha	100.0%	100.0%

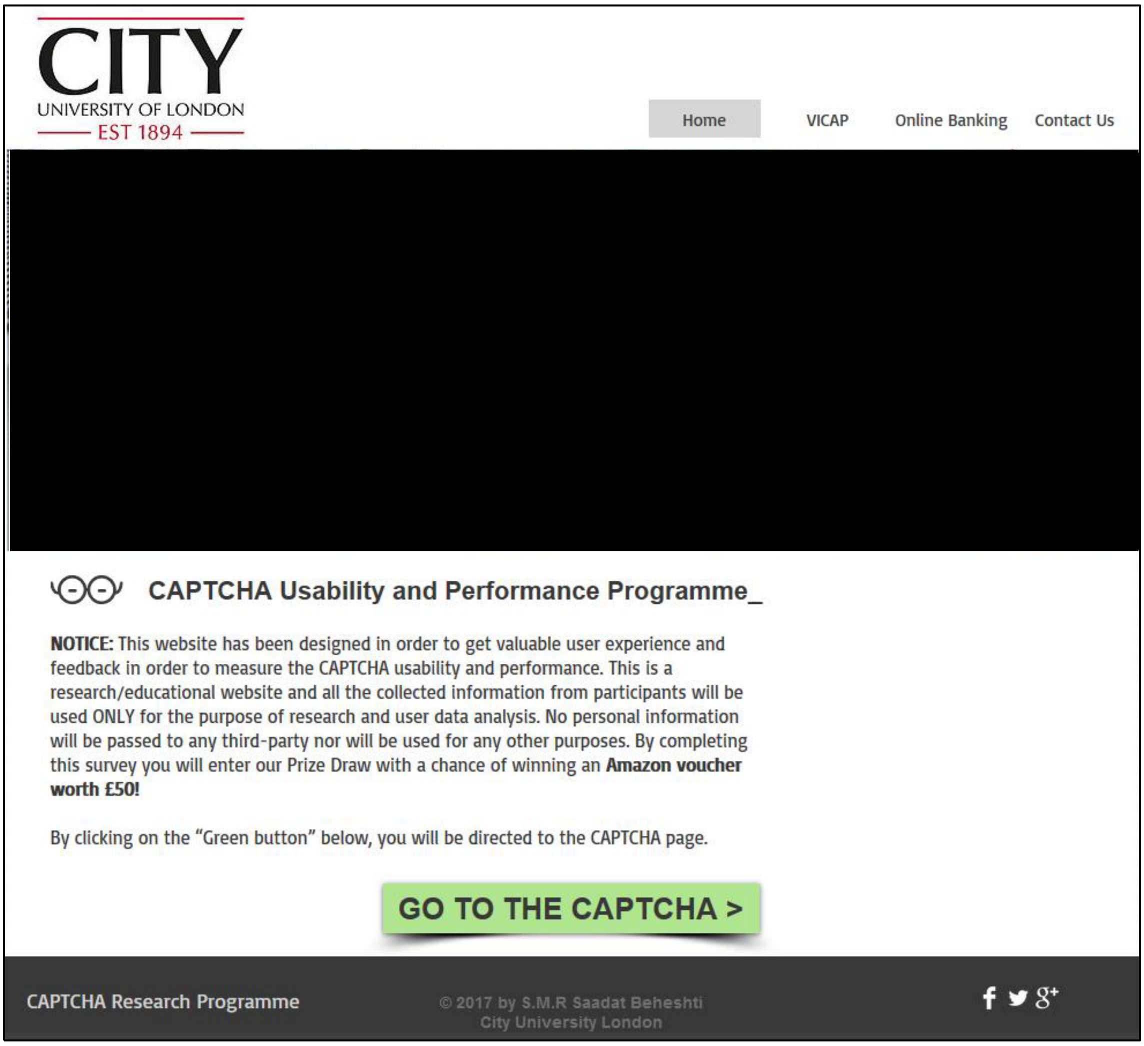
5.3.2 Experiment 2: Human Recognition Success Rate (HRSR)

In previous tests on our proposed CAPTCHA model, we worked out the highest threshold value that the ORO parameter could be in order to give a CRSR of 0%. As already acknowledged, any CAPTCHA model should satisfy two conditions to be considered valid: namely, be too difficult or impossible for computers to break, and very easy for humans to solve. Therefore, it was important to carry out another set of experiments on human users in order to find out how high the ORO parameter level could before the CAPTCHA test became too difficult for a human to solve. The aim of running these experiments was to find the optimal level for the ORO parameter, one that gives the highest clarity to a human user, but the lowest recognition success rate for computer programmes.

To run the second experiment, the CAPTCHA User Evaluation Website[1] was designed to enable users to participate in the VICAP model evaluation programme and provide their feedback regarding usability. In total, 150 participants from different age groups and backgrounds participated in the experiment through this website. The website is designed in such a way that every time a user visits the website, a new set of random

[1] The CAPTCHA User Evaluation Website is accessible at *http://mrbeheshti2.wixsite.com/captcha-project*

characters and numbers is generated and displayed to them by recording their IP address. Figure 5.16 displays a screenshot of the CAPTCHA User Evaluation Website.



[redacted]

CAPTCHA Usability and Performance Programme_

NOTICE: This website has been designed in order to get valuable user experience and feedback in order to measure the CAPTCHA usability and performance. This is a research/educational website and all the collected information from participants will be used ONLY for the purpose of research and user data analysis. No personal information will be passed to any third-party nor will be used for any other purposes. By completing this survey you will enter our Prize Draw with a chance of winning an **Amazon voucher worth £50!**

By clicking on the "Green button" below, you will be directed to the CAPTCHA page.

Figure 5.16: A screenshot of the CAPTCHA evaluation website.

Feedback from the 150 participants was collected and the average value for each grade of the ORO parameter was calculated based on the ability of the users to see and recognise the CAPTCHA challenge. Figure 5.9 shows a comparison of 11 different ORO parameters and their associated HRSR. As it can be seen from the table, during the test the ORO equalled 0%. Where there are no original frames presented, no useful information could be obtained by the users. Consequently, as we can see from the results, the HRSR is also equal to 0%. By increasing the value of the ORO parameter, the recognition success rate for the users also increases rapidly.

However, as the experimental results confirm, the sophisticated recognition abilities of the HVS react more sharply and accurately than those of computer recognition programmes. As it can be deduced from Figure 5.9, for ORO < 20%, the HRSR is almost equal to 0%. This means that for ORO < 20%, there is not enough sufficient information about the object and it is difficult for human users to see or recognise it. However, as the results from Table 5.6 shows, the first big jump in terms of the HRSR is at ORO = 20%, which gives a success rate of 65%. The second experiment is similar to the first. The threshold value is also measured at ORO equally 20%, and there is a similar significant jump in terms of the HRSR.

Table 5.6: A comparison of different ORO frame rate parameters and the effect these have on the HRSR.

ORO Parameters	Human Recognition Success Rate
0%	0%
10%	0%
20%	65%
30%	80%
40%	100%
50%	100%
60%	100%
70%	100%
80%	100%
90%	100%
100%	100%

In the second set of experiments, we aimed to determine whether the sharp jump in the HRSR in the range of ORO values equalling between 10% and 20% is instantaneous or else follows a gradual increase. To test this, the VICAP evaluation website was used. Over 50 participants completed this CAPTCHA user experience evaluation test. The feedback provided by the participants was collected and the output results will be analysed and discussed shortly. As shown in Table 5.7, the final output results from over 50 participants confirm that the HRSRs for ORO values of up to 19% was equal to 0%. This means that no participants could recognise the entire CAPTCHA string when the ORO parameter is 19% or less. At an ORO rate of 20%, the HRSR suddenly jumps to

60%, meaning some CAPTCHAs were deciphered completely by most participants. As it can be observed from these new output results, the new HRSR is very close to the previous experiments where the HRSR was measured at 65% when the ORO equalled 20%. This is because, in the second set of user experiments, the total number of participants was lower than the total number of participants in the first experiment. Additionally, since the output results are dependent on human users, whose abilities range widely, a slight difference in recognition success rates between the two experiments is expected.

Table 5.7: Changes in the HRSR for the entire CAPTCHA string when varying the ORO parameters.

ORO Parameters	HRSR
10%	0%
11%	0%
12%	0%
13%	0%
14%	0%
15%	0%
16%	0%
17%	0%
18%	0%
19%	0%
20%	60%

In order to identify the reason for this big jump in terms of the HRSR, we analysed the ‘intermediate’ output results of individual CAPTCHA character recognition rates and their ability to recognise the entire CAPTCHA string. These results are shown in Table 5.8.

Table 5.8: Changes in the human recognition success rate for partial CAPTCHA character recognition when varying the ORO parameters.

ORO Parameters	HRSR
10%	0%
11%	0%
12%	0%
13%	4%
14%	8%
15%	12%
16%	20%
17%	28%
18%	40%
19%	56%
20%	84%

Up to an ORO level of 12%, the HRSR is 0%. However, following this level, there is a nearly linear increase for an ORO rate of up to 17%. For values higher than 18%, human users could recognise most individual CAPTCHA characters with increasing accuracy, rising to 84% at an ORO of 20%. This, in turn, translates to the entire CAPTCHA string becoming more readable to more users. The output results for ORO equalling 20% in Table 5.7 indicate that three out of five users could correctly recognise the entire CAPTCHA string (the HRSR equals 60%).

5.3.3 ORO Analytical Comparison

Our proposed CAPTCHA model is designed specifically to work in collaboration with the HVS. Therefore, the expectation is to get better and more accurate results for humans rather than computers. Figure 5.17 shows a comparison in terms of the recognition success rates for both human users and computer recognition programmes versus the different ratios of the ORO parameter. As the graph confirms, the HRSR rises earlier and faster than the CRSR. In order to make the results clearer, we have added a new plot in Figure 5.17, which demonstrates the CRSR ‘Per Single Frame.’ This allows a comparison to be made with the multi-frame sequence recognition for both computer programmes and humans.

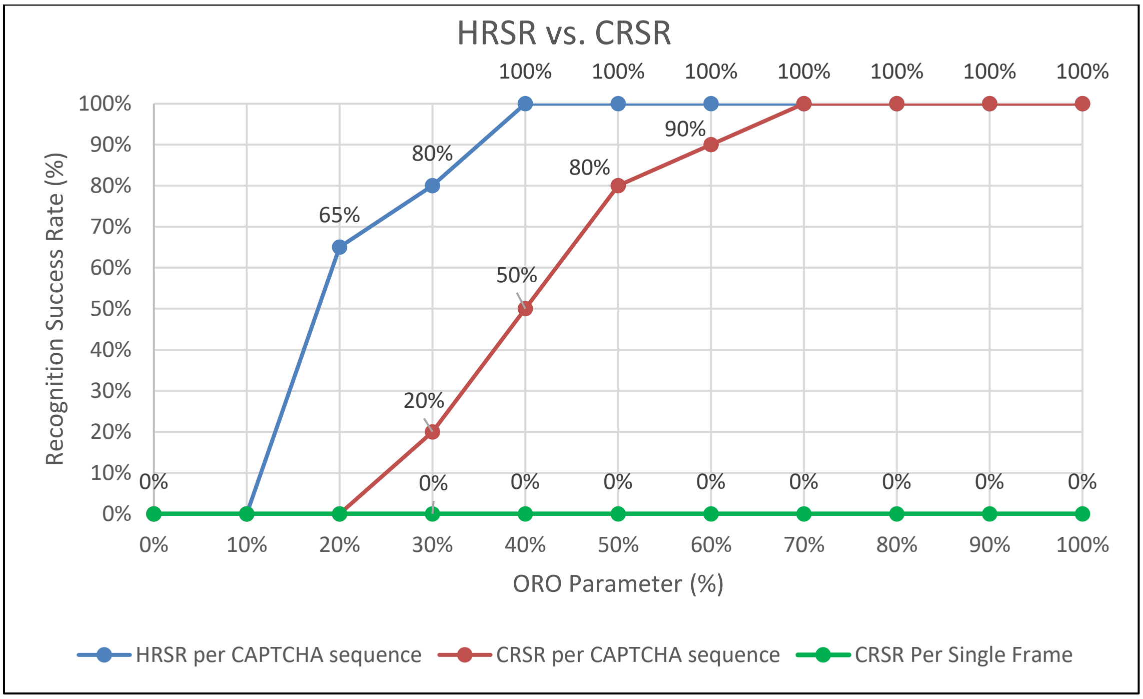


Figure 5.17: A comparison of CRSR vs. HRSR per different ORO values. As the graph shows, human recognition stands above computer recognition success rate. The ideal situation is measured at 40% where HRSR is at 100% but CRSR is at 50% for a multi-frame and 0% for a single frame.

Our new CAPTCHA model was tested in two scenarios. The first scenario considers the traditional setup of a computer attack, where a single frame of the CAPTCHA is seized and passed on to OCR software for recognition. The second case, implemented through our CTA, uses prior knowledge of the CAPTCHA design. A number of frames are individually captured and superimposed (or integrated) to generate output images, as the ones presented in Figure 5.8 to Figure 5.12. The second scenario is biased because it also requires prior knowledge of the time interval (ISI) to be used in the integration process. When the time interval is set to a value higher than the optimal ISI, there is insufficient information to complete the CAPTCHA string. When the time interval for integration is set to a value lower than the optimal one, the CAPTCHA image is saturated due to the uniform nature of the noise process used for the background.

In order to ensure the usability of the proposed CAPTCHA model, we set the threshold for the ORO parameter to 40%. This ensures that our CAPTCHA strings are recognised by human observers at a rate of 100%. In turn, when it comes to examining the robustness of VICAP to computer programme attacks, we can observe that, for the traditional case of OCR recognition software based on a single-frame scenario, the CRSR is about 0%, whilst in the case of a multi-frame scenario, the CRSR can increase to up to 50%. In the unlikely scenario of an advanced OCR software attack comprising of frame integration over an optimal time interval, the robustness of the VICAP model for the multi-frame sequence reduces to 50%. However, we must stress that this latter scenario is unfairly biased because it is not supported by the present capabilities of state-of-the-art OCR software.

We need to emphasise here that every single Original VICAP frame is made up of random pixels (or dots) with a rate of 25% for the OSR and 15% for the BNR, which presents only partial information about the hidden object. Therefore, as the table below confirms, every individual frame contains no meaningful information but a random noise (as was shown in Figure 4.14). Therefore, the average value of the CRSR per single frame will stay at 0%, regardless of the value of the ORO parameter because there is no meaningful information presented to the OCR programme. As the results in Table 5.9 confirm, 20 randomly selected output VICAP images for each ORO category have been captured and fed into an OCR programme directly without applying a superimposing process. In total, 220 VICAP images have been tested and the final output CRSR per single image has been presented in the table below.

Table 5.9: The average value of the CRSR for 220 VICAP images analysed individually by an OCR programme.

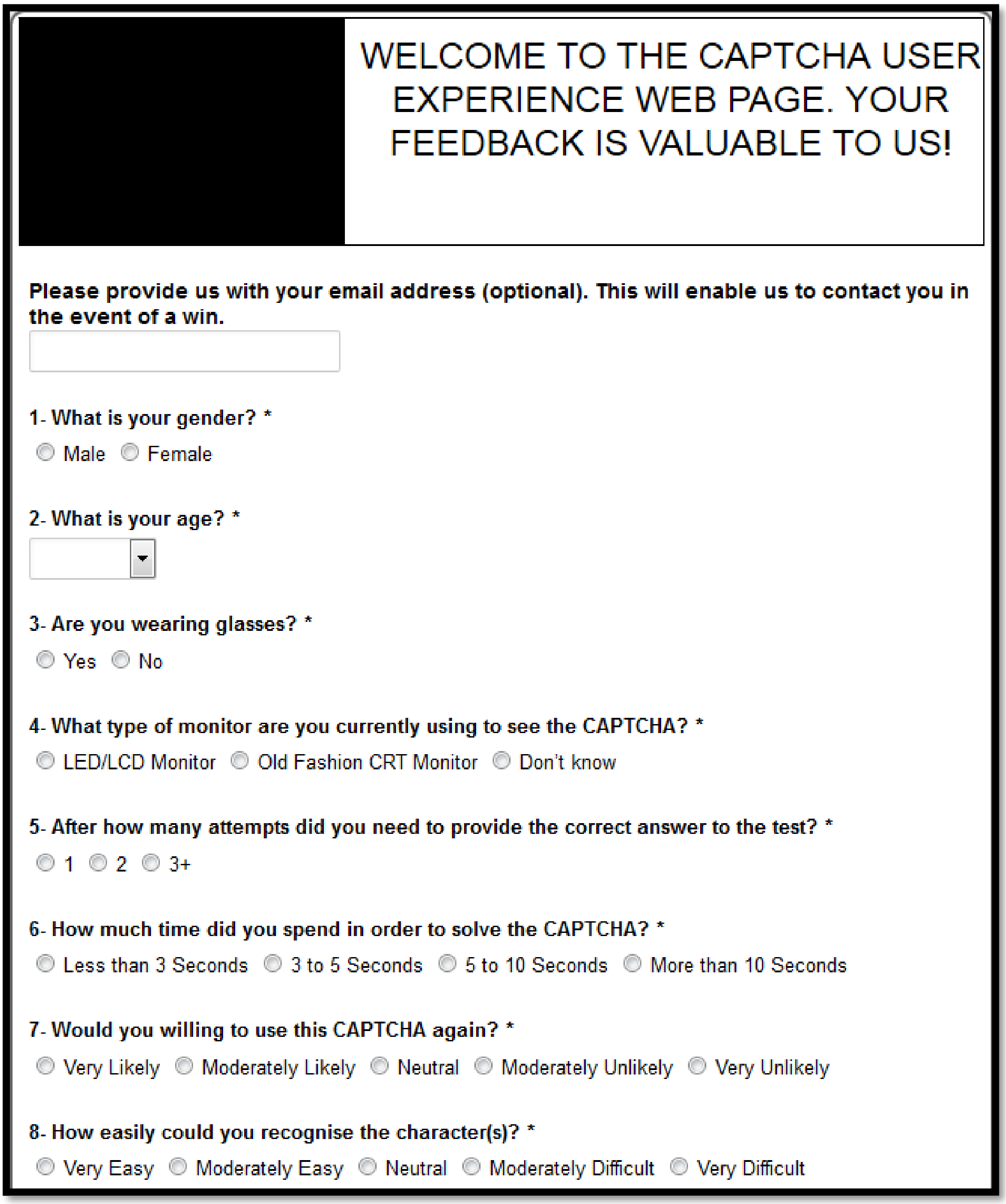
ORO Parameters	CRSR Per Single Frame
0%	0%
10%	0%
20%	0%
30%	0%
40%	0%
50%	0%
60%	0%
70%	0%
80%	0%
90%	0%
100%	0%

To conclude, the only way to make the model understandable for an OCR programme is to superimpose a number of frames with the conditions that have already been explained briefly and feed the single output rendered image into the OCR software. ORO parameters will control the probability of the ‘original’ frames over ‘random’ frames. Thus, by setting the ORO parameter to 0%, there will be no original frames present in the sequence of output frames and therefore no meaningful information will be presented apart from pure random noise. Analysing every individual random frame will produce a CRSR of 0%. By setting the ORO parameter to 100%, there will only be original frames presented in the video sequence. As previously discussed, every single frame is rendered in such a way as to have no useful information on its own, but a combination of frames will create the effect of POV in the human brain. Therefore, similarly analysing individual frames will produce a CRSR of 0%.

5.4 VICAP v.2 Usability and Performance Experimental Results

The aim of this research project is to investigate various aspects that can affect the usability and performance levels of CAPTCHAs as it was explained in [66], as well as measure the usability and performance of our own proposed CAPTCHA model quantitatively and qualitatively by comparing it to current CAPTCHA models. In order to achieve this goal, there needs to be a good understanding of the various issues that can affect the usability and performance of CAPTCHAs. These have already been discussed in detail in section 2.10.

In this section, I will demonstrate the impact of those issues in terms of usability and performance using real CAPTCHA examples. In order to gain valuable user feedbacks, a comprehensive user survey needed to be designed. Regardless of the application, the main aim of any survey is to design a useful questionnaire that can probe deeply and produce meaningful data. The design of such a questionnaire is a considerable challenge because it is necessary to measure user experience both qualitatively and quantitatively. In total, 13 questions were designed carefully in order to address a variety of usability issues, as discussed in previous chapters of this thesis. Some of the questions were designed quantitatively, such as age, whether or not the user wears glasses, time to solve, and correct or incorrect response etc. Other questions focused on qualitative elements, such as user participation, character recognition levels, CAPTCHA difficulty levels, etc. Figure 5.18 displays a partial screenshot of the questionnaire we designed. The full online questionnaire form is accessible at [67].



[redacted]

WELCOME TO THE CAPTCHA USER EXPERIENCE WEB PAGE. YOUR FEEDBACK IS VALUABLE TO US!

Please provide us with your email address (optional). This will enable us to contact you in the event of a win.

1- What is your gender? *

○ Male ○ Female

2- What is your age? *

3- Are you wearing glasses? *

○ Yes ○ No

4- What type of monitor are you currently using to see the CAPTCHA? *

○ LED/LCD Monitor ○ Old Fashion CRT Monitor ○ Don't know

5- After how many attempts did you need to provide the correct answer to the test? *

○ 1 ○ 2 ○ 3+

6- How much time did you spend in order to solve the CAPTCHA? *

○ Less than 3 Seconds ○ 3 to 5 Seconds ○ 5 to 10 Seconds ○ More than 10 Seconds

7- Would you willing to use this CAPTCHA again? *

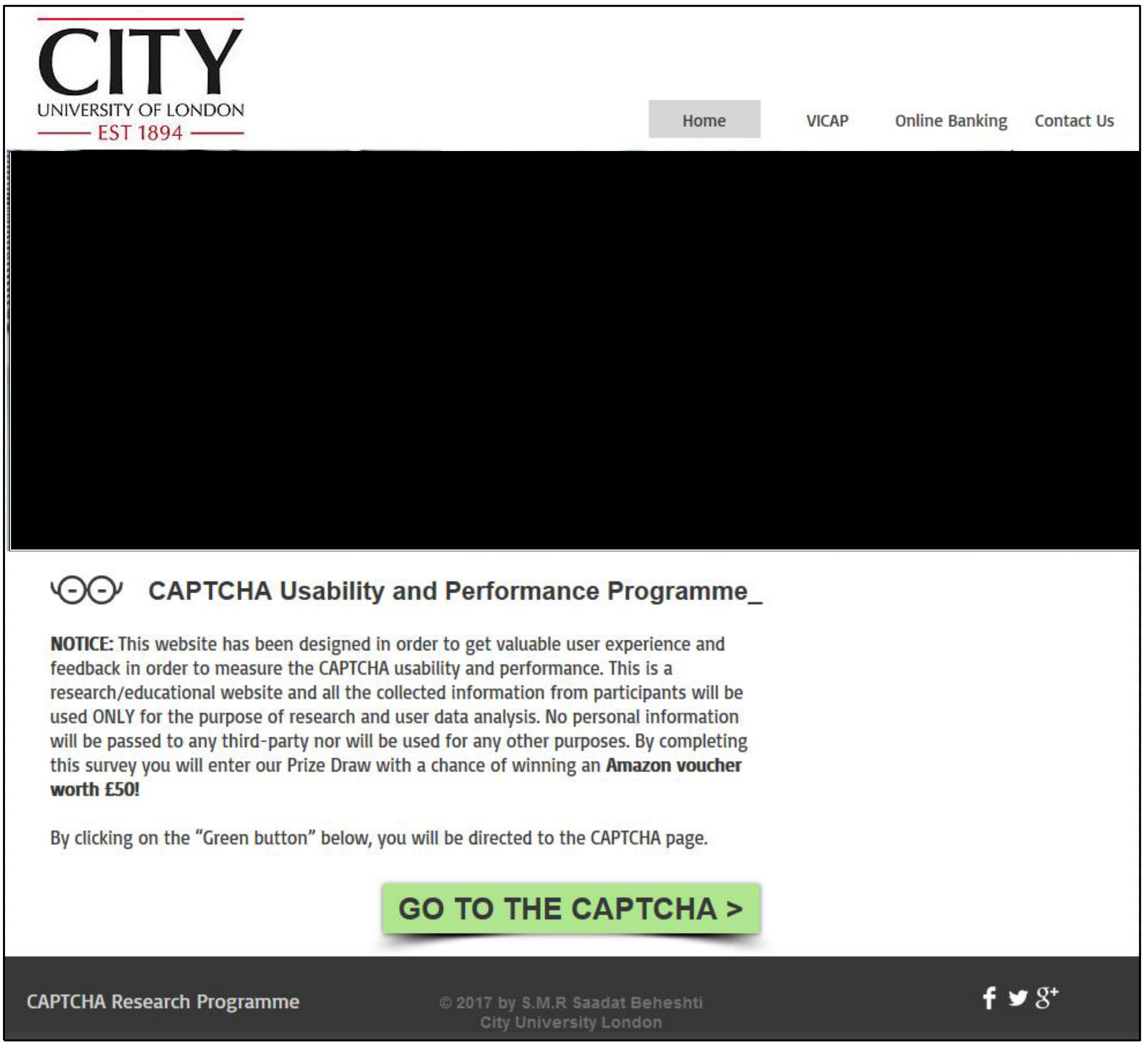
○ Very Likely ○ Moderately Likely ○ Neutral ○ Moderately Unlikely ○ Very Unlikely

8- How easily could you recognise the character(s)? *

○ Very Easy ○ Moderately Easy ○ Neutral ○ Moderately Difficult ○ Very Difficult

Figure 5.18: A partial screenshot of a questionnaire designed for our CAPTCHA user experience programme.

In order to conduct these experiments, a website has been developed entitled CAPTCHA User Experience and Performance (CUEP). This website was developed to compare our proposed CAPTCHA model in terms of usability and performance with one of the most popular current CAPTCHA models, known as ReCAPTCHA. The CUEP website allows a user to provide feedback for every CAPTCHA model via an online survey form. The CUEP website is accessible at [68] and a screenshot of the homepage is displayed in Figure 5.19.



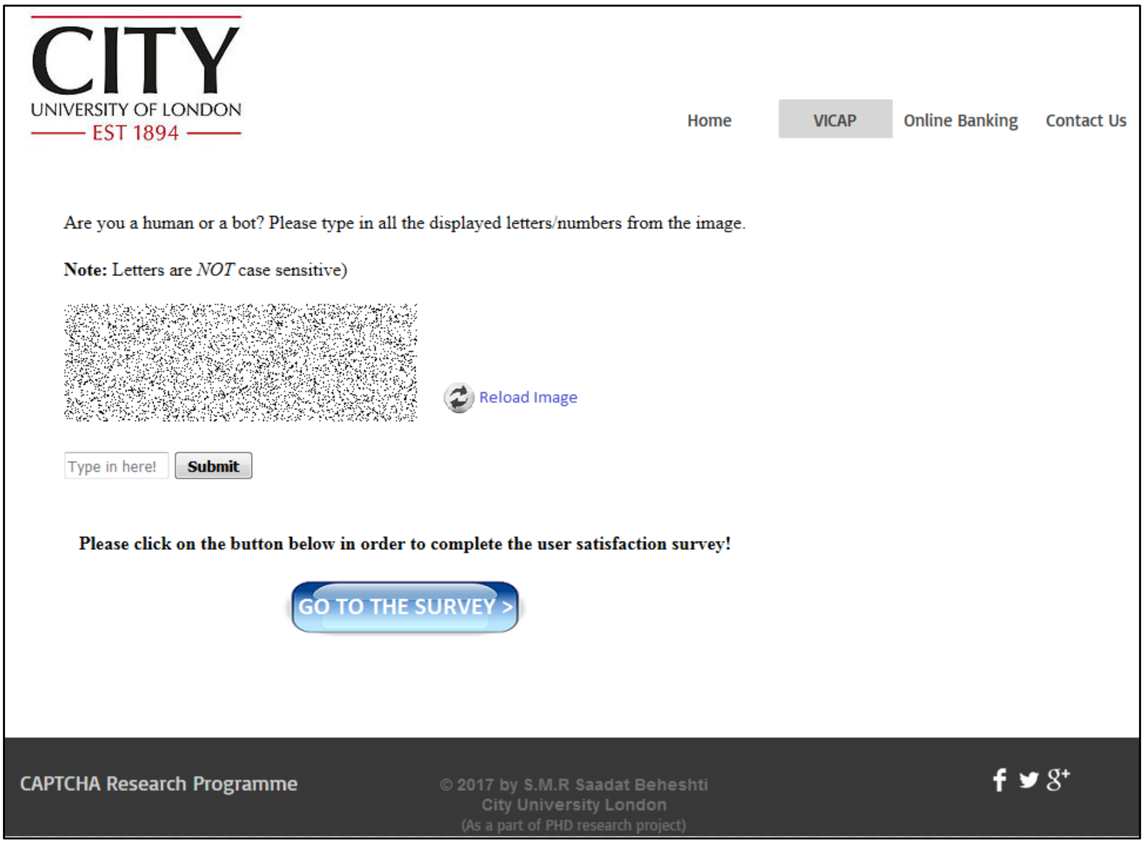
CAPTCHA Usability and Performance Programme_

NOTICE: This website has been designed in order to get valuable user experience and feedback in order to measure the CAPTCHA usability and performance. This is a research/educational website and all the collected information from participants will be used ONLY for the purpose of research and user data analysis. No personal information will be passed to any third-party nor will be used for any other purposes. By completing this survey you will enter our Prize Draw with a chance of winning an **Amazon voucher worth £50!**

By clicking on the "Green button" below, you will be directed to the CAPTCHA page.

Figure 5.19: The CUEP homepage view.

As we can see from the above image, all participants will be given instructions on how to use the CUEP website. By clicking on the green button, participants will be directed to the next page where he or she can take part in the CAPTCHA challenge. On the second webpage, as shown in Figure 5.20, the users will be invited to complete both our proposed CAPTCHA model and the ReCAPTCHA model. After completing both CAPTCHA models, the users will be offered to take the survey by clicking on the blue button at the bottom of the page, as shown in Figure 5.20. Due to space limitation in this thesis, it is not possible to demonstrate the effect of POV here because a screenshot of our CAPTCHA would only produce an image containing random dots. Thus, the real effect must be experienced in practice.



Are you a human or a bot? Please type in all the displayed letters/numbers from the image.

Note: Letters are *NOT* case sensitive)

Please click on the button below in order to complete the user satisfaction survey!

© 2017 by S.M.R Saadat Beheshti
City University London
(As a part of PHD research project)

Figure 5.20: The participant is presented with the VICAP model.

5.4.1 Comparison of VICAP and Current CAPTCHA Model in terms of Usability and Performance

In our online questionnaire, participants were required to complete a total of 13 different questions. 100 participants from a wide range of backgrounds, ethnic groups, and age categories contributed to this user experience programme. In this section, we have provided a number of different graphs displaying all participants’ responses. Some graphs represent the quantitative data gathered, and some represent the qualitative data gathered. For instance, as it can be observed in Figure 5.21, the vast majority of the participants were male (approximately 71%) and only 29% of the participants were female.

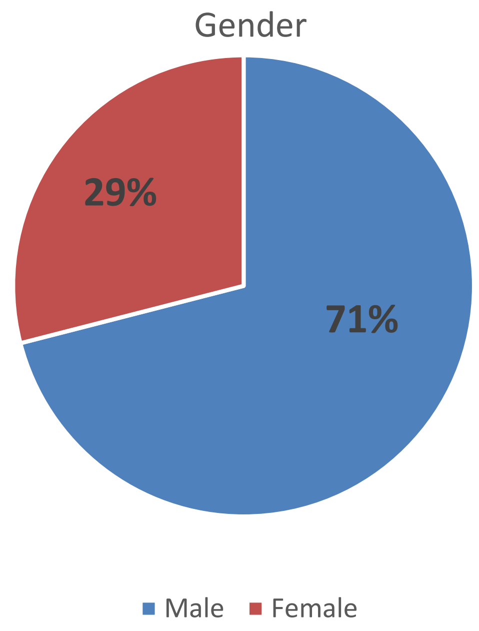


Figure 5.21: Gender distribution of participants.

As it can be observed from the results, some of the questions were designed to measure quantitative information about the participants, such as Figure 5.21, which identifies their gender. Figure 5.22 is also quantitative because it represents the age categories of participants. As seen in this graph, the vast majority of participants were aged between 21 and 30 years old, which indicate that they were mostly students. A smaller portion of the participants was aged between 31 and 40 years old, which indicate that they were mostly university staff. It is important to know which groups of people solve CAPTCHAs most often in order to create a better and more robust model because some CAPTCHAs might be found to be more difficult for certain age groups.

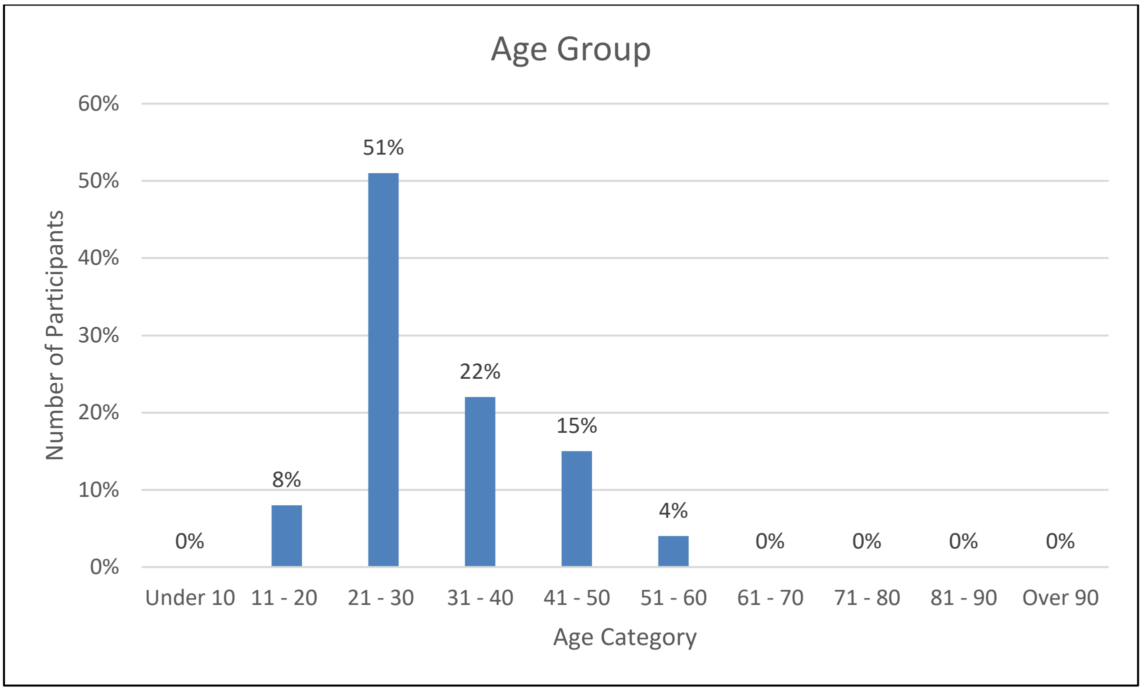


Figure 5.22: Age distribution of participants

Figure 5.23 shows what fraction of the participants wear glasses. This is another quantitative graph and shows that 73% of the participants do not wear glasses, whilst 27% of the participants wear glasses.

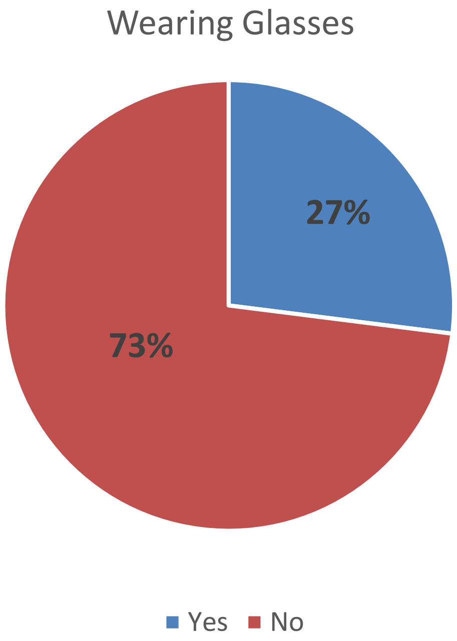


Figure 5.23: Number of participants who wear glasses.

Another essential factor for us to identify is what type of monitor participants were using. Our proposed CAPTCHA model is based on POV, so the quality of pixels is important, and a participant’s monitor type may affect the quality of the pixels. Since the pixel rates would be different depending on the monitor’s screen refresh rate, it is important to know what types of displays the majority of users are using in order to increase the usability of our CAPTCHA model. Figure 5.24 shows the different screen types of the participants. As we can see from the graph, 94% of participants are using LED/LCD monitors, which use the standard refresh rate.

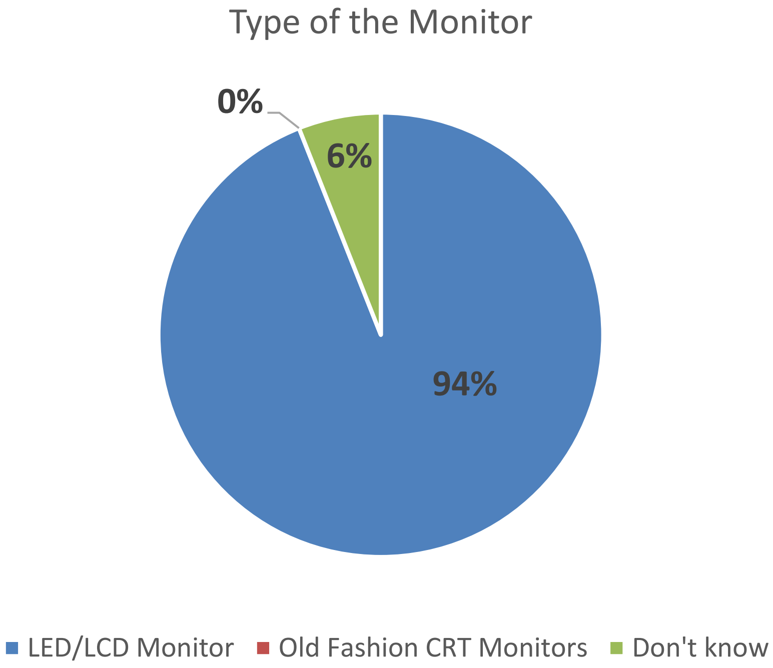


Figure 5.24: Types of monitor used by participants.

Figure 5.25 is also quantitative as it represents the number of attempts the users made in order to provide a correct response to the CAPTCHA challenge. As can be seen from the graph, two CAPTCHA models have been compared and the final output results are presented. The blue colour represents the current ReCAPTCHA model and the red colour represents our proposed VICAP model. As we can see from Figure 5.25, 98% of participants could solve and recognise our proposed VICAP model on the first attempt, while 61% of the participants could solve the current ReCAPTCHA model on the first attempt. Similarly, 28% of participants could solve the ReCAPTCHA model after two attempts, while only 2% of participants could solve our proposed VICAP model after two attempts. Therefore, we can conclude that our proposed VICAP model can be solved faster and more easily than ReCAPTCHA, with a performance improvement of more than 37% in the first attempt. This element can be measured quantitatively.

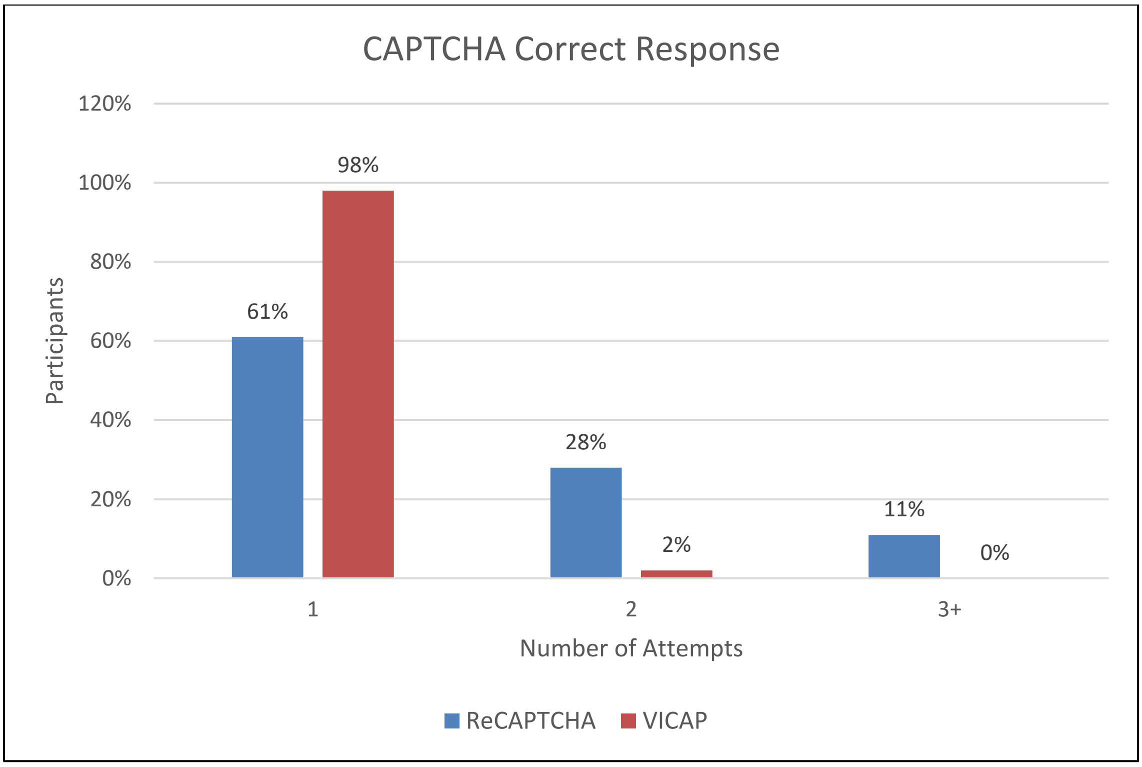


Figure 5.25: The number of attempts required by participants in order to provide a correct response to the CAPTCHA challenge.

Another important quantitative measurement to consider is the time it took participants to solve the CAPTCHA test. This refers to how quickly a user is able to answer the CAPTCHA correctly, which would affect the usability of a CAPTCHA. As we can see from Figure 5.26, the ‘time to solve’ has been measured in seconds.

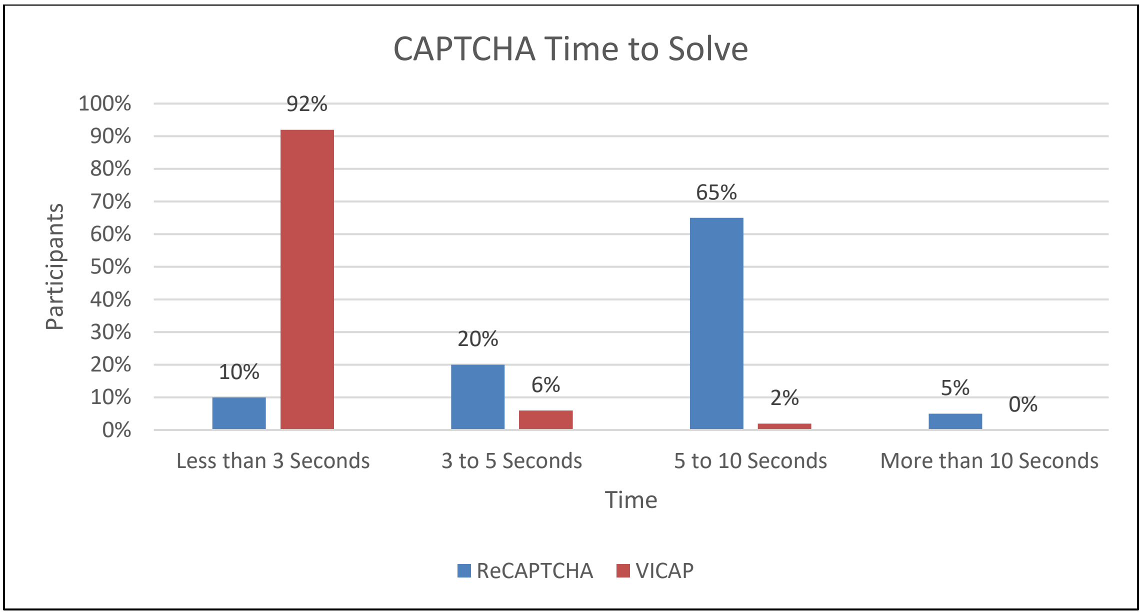


Figure 5.26: The different time intervals of participants to solve the CAPTCHA test.

As Figure 5.26 shows, the average time it took for most participants to solve the ReCAPTCHA test is measured at 5 to 10 seconds, while the time it took for the vast majority of participants to solve the VICAP test was less than 3 seconds. These results confirm that our proposed VICAP model is faster and has a higher performance rate than other current CAPTCHA models.

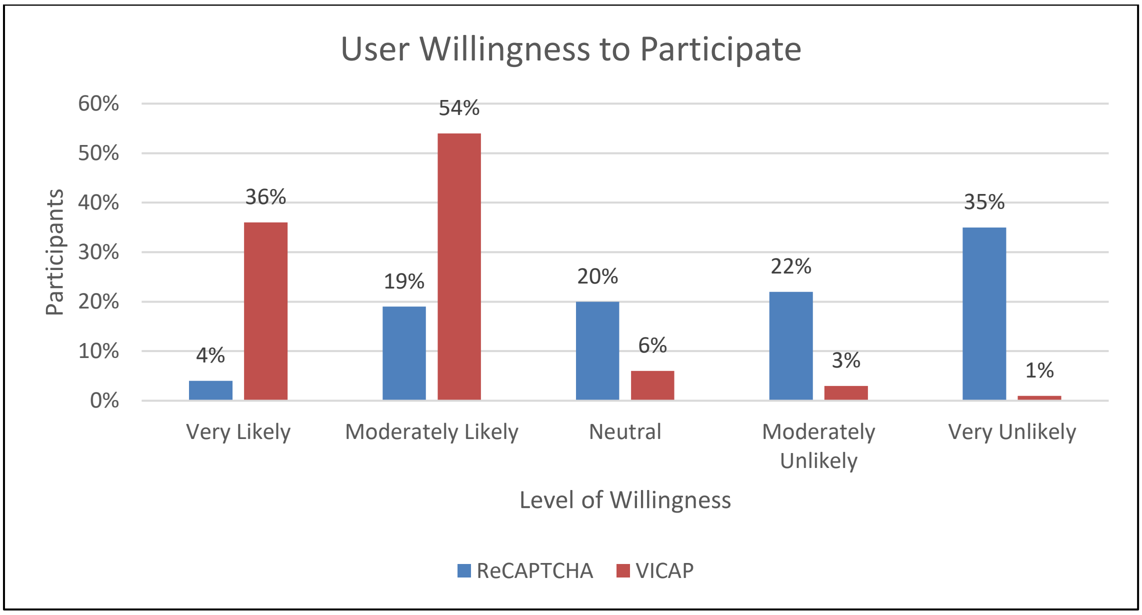


Figure 5.27: The level of willingness of participants to take part in future challenges.

Figure 5.27 is qualitative data and shows how likely participants are to use this CAPTCHA model again in the future. As the feedback reveals, 35% of participants voted 'very unlikely' to use the ReCAPTCHA model again in the future and 22% of the participants voted 'moderately unlikely.' On the other hand, 54% of participants voted 'moderately likely' to use the VICAP model again in the future, and 36% voted 'very likely.' Again, this data shows that VICAP is more appealing and user-friendly than other current models.

Figure 5.28 displays the difficulty level participants experienced for each CAPTCHA model, which is another qualitative element. This measures how hard each participant found the challenge to be. As Figure 5.28 reveals, 45% of participants rated the ReCAPTCHA model as 'moderately difficult,' while a substantial 65% of participants rated the VICAP model as 'very easy.' From these results, we can appreciate that our proposed CAPTCHA model is far easier to complete than the current ReCAPTCHA model.

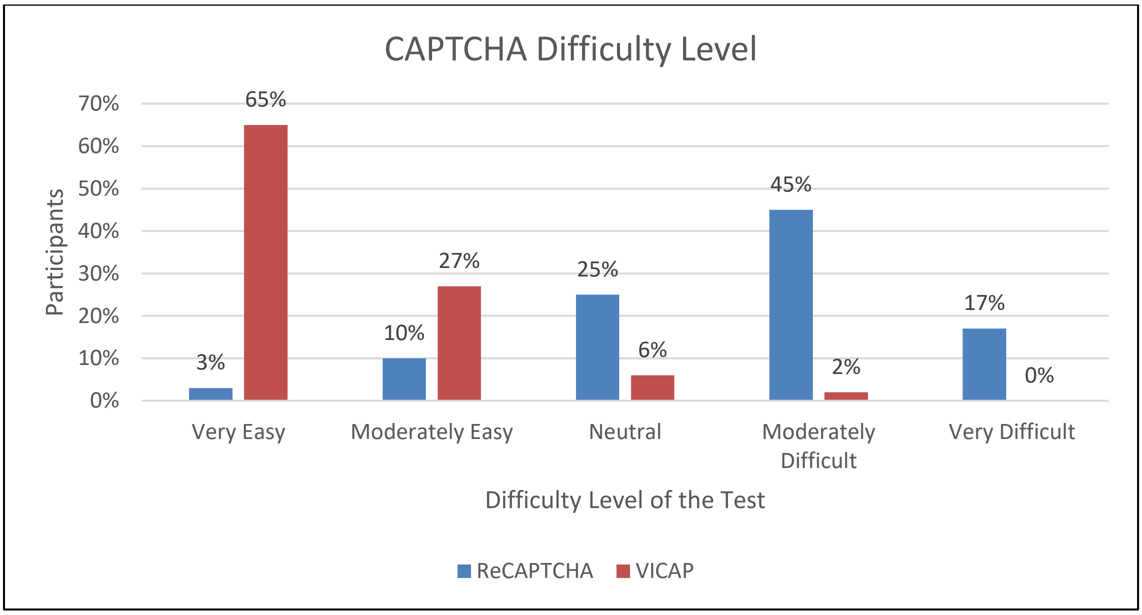


Figure 5.28: The level of difficulty in terms of human character recognition.

One of the most important factors that can improve the usability of a CAPTCHA is the clarity level of the characters and letters, which can also be measured qualitatively. As it is shown in Figure 5.29, the ambiguity level of the characters can affect the usability of the CAPTCHA dramatically. 56% of participants labelled the VICAP characters as being 'moderately clear' and 35% of participants rated them as being 'very clear.' However, in the case of the current ReCAPTCHA model, 47% of participants rated the characters as being 'moderately unclear' and 35% of participants rated them as being 'very unclear.' These results clearly confirm that the characters of our proposed CAPTCHA model will be more readable and more recognisable for the user, making it more user-friendly.

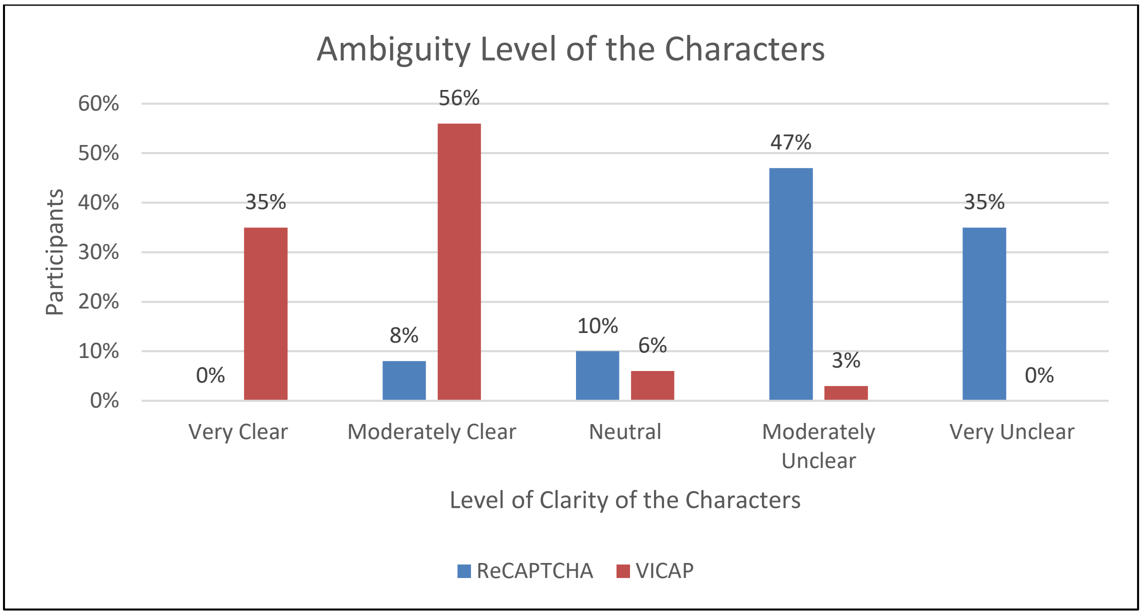


Figure 5.29: The ambiguity level of the characters as experienced by the participants.

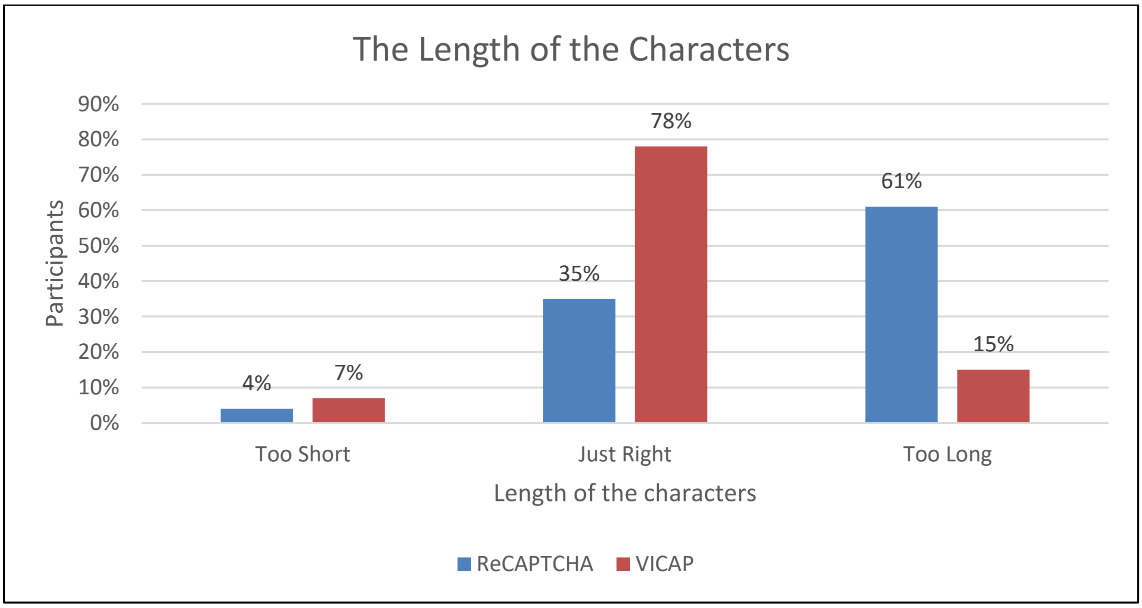


Figure 5.30: The length of the characters in each CAPTCHA challenge.

Figure 5.30 represents another qualitative element that can affect the usability of CAPTCHAs, which is the length of the characters. The number of characters and numbers can have an impact on the usability and performance of CAPTCHA models because having very long CAPTCHAs takes more time and energy for users to decipher. On the other hand, having too few characters can affect the security of a CAPTCHA as it would make it easier for an OCR programme to recognise. As Figure 5.30 shows, around 61% of participants labelled the ReCAPTCHA characters as 'too long' and only 35% labelled them as 'just right.' Yet, 78% of participants labelled the VICAP characters as 'just right' and only a fraction of participants labelled them as 'too long' (15%). What we can understand from these results is that the current ReCAPTCHA model is too difficult for the majority of users, while most users would be more comfortable with our proposed VICAP model.

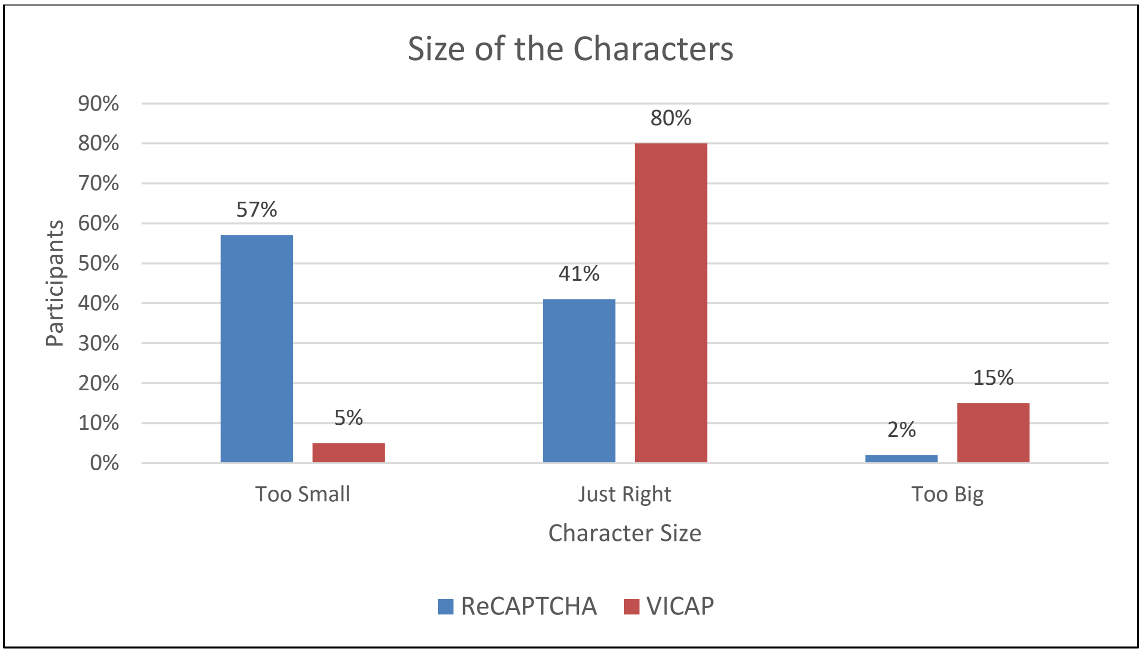


Figure 5.31: User experience in terms of the size of the characters in each CAPTCHA challenge.

The size of the characters of a CAPTCHA challenge is another qualitative element that is critical to the usability and performance of the test. Very small characters will be more difficult for users to read, but overly big characters may reduce the security of the CAPTCHA because bigger characters are more easily recognisable by OCR software. Figure 5.31 compares how participants reacted to the character sizes of both the current ReCAPTCHA model and our proposed VICAP model. As we can see from the results, 57% of participants rated the ReCAPTCHA characters as ‘too small’ and 41% rated them as ‘just right.’ On the other hand, 80% of participants rated the VICAP characters are being ‘just right,’ which is a substantial portion. This also shows that our proposed CAPTCHA model is easier to use, which has a positive impact on its usability and performance.

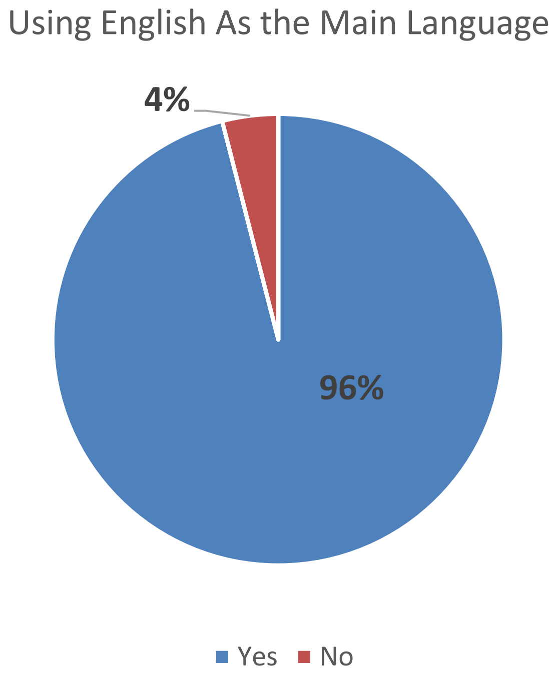


Figure 5.32: The distribution of English speaking participants.

As it has been discussed in previous chapters of this thesis, one factor that can directly affect the usability of a CAPTCHA is the language. Since text-based CAPTCHAs use written text, it is necessary to know the language of the location where the CAPTCHA is being solved. As Figure 5.32 shows, 96% of participants speak English as their main language, and only 4% of participants do not speak English as their main language. Use of different languages is a quantitative element, and we would be able to measure the total number of people who speak a particular language in each location, which would then dictate the design of our CAPTCHA test for that particular group.

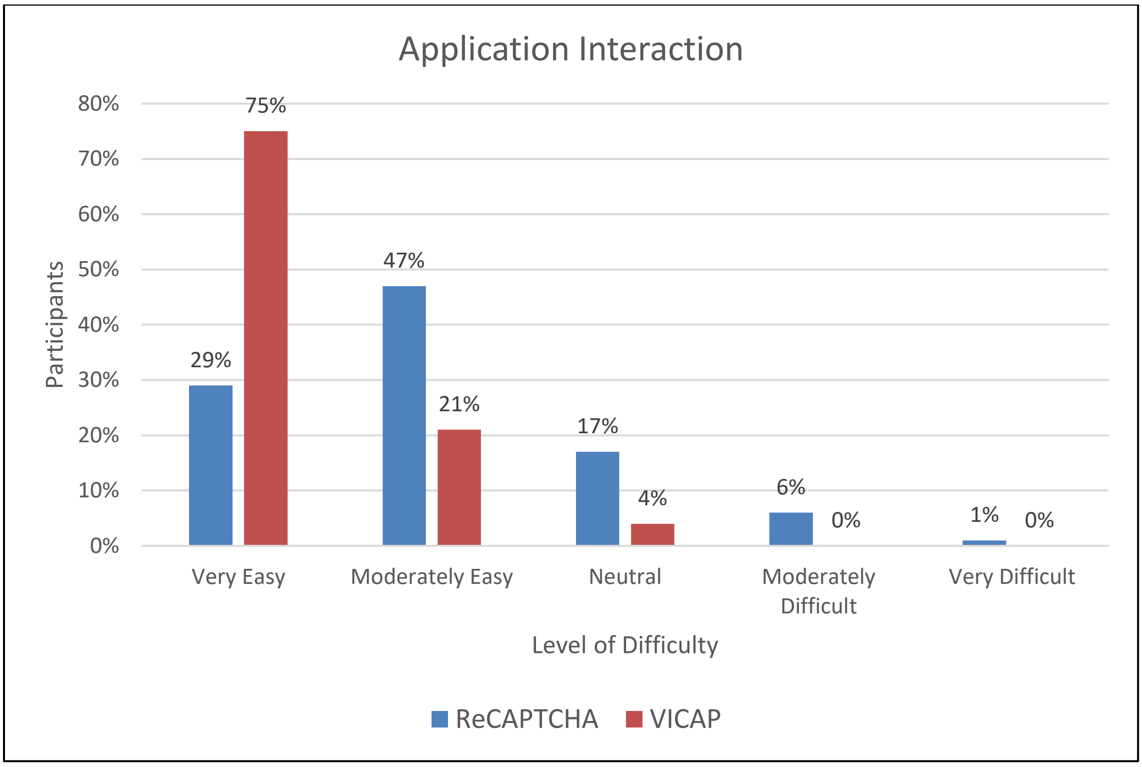


Figure 5.33: The user experience in terms of the difficulty level of the CAPTCHA application.

Finally, integrating our CAPTCHA application into a third-party webpage is a very important and complicated job. A CAPTCHA should be integrated into a webpage or online form, making it easily accessible for the end user to access. For this reason, we included the factor of 'application interaction' in our online survey, which can be measured as a qualitative element. The application interaction level measures how accessible a CAPTCHA challenge is to the user. As shown in Figure 5.33, 47% of participants rated the application interaction of the current ReCAPTCHA model as 'moderately easy,' and only 29% of participants labelled it as 'very easy.' Conversely, 75% of participants rated the application interaction of the VICAP model as 'very easy' and 21% of participants rated it as 'moderately easy.' This shows that our VICAP model has the better performance rating in terms of CAPTCHA application interaction.

5.4.2 Diversions in Human Perception

The recognition levels and robustness of the proposed VICAP model have now been measured and analysed for both human users and computer recognition programmes. Hence, we would now like to discuss whether there were any exceptional cases that may affect the sophisticated HVS, and cause it to be less reliable than a computer recognition programme.

We have interpreted the definition of what we will call ‘diversion in human perception’ in two ways. Firstly, we will analyse the HRSR in the context of the best and worst scenarios, as recommended by the reviewer, and compare these to computer recognition programmes. To support this, we will present samples from previous experimental results. Secondly, we will comment on the influence of random frames on the proposed CAPTCHA design and discuss the effect this has on visual recognition.

Our experimental results confirm that the HVS successfully interfaces with the proposed CAPTCHA model, whereas traditional computer recognition programmes do not. Specifically, human users can recognise our CAPTCHA characters with an HRSR of 100% with ORO rates of 40% or higher, while equal recognition rates are only possible for computer programmes with ORO rates of 70% or higher. In total, we have conducted over 200 experiments to analyse CRSRs, and over 150 experiments involving more than 50 human participants in order to measure HRSRs. Table 5.10 summarizes the best and worst-case scenarios for CRSRs and HRSRs respectively when varying the ORO levels. In this table, 0% represents the absolute worst and 100% represents the absolute best recognition rate. As it can be observed, there are no instances where the best case HRSR is lower than the best case CRSR. The HRSR is either equal to, or higher than, the CRSR. In summary, our proposed CAPTCHA design, based on the properties of the HVS and IM, guarantees that human recognition rates outperform current computer programme recognition rates.

Table 5.10: Best and worst-case scenarios for recognition success rates for human users and computer programmes when varying ORO levels.

ORO	HRSR-Best Case	HRSR-Worst Case	CRSR-Best Case	CRSR-Worst Case
0%	0%	100%	0%	100%
10%	0%	100%	0%	100%
20%	60%	40%	0%	100%
30%	80%	20%	20%	80%
40%	100%	0%	50%	50%
50%	100%	0%	80%	20%
60%	100%	0%	90%	10%
70%	100%	0%	100%	0%
80%	100%	0%	100%	0%
90%	100%	0%	100%	0%
100%	100%	0%	100%	0%

The second aspect that needs to be discussed is the influence of a sudden stimulus on a human user’s perception when attempting CAPTCHA recognition. Indeed, it is well known that there are neural mechanisms by which the brain detects and responds to novelty (i.e. the presence of a sudden stimulus) [69]. Examples of these stimuli could be a person suddenly walking into the user’s environment or a telephone suddenly ringing, which may temporarily require the user’s attention. This could affect the ability of the user to recognise CAPTCHAs, particularly in regard to the time it takes to solve the challenge. In some respects, the proposed CAPTCHA model can be considered as also using ‘sudden’ stimuli because the sequence of the CAPTCHA’s original frames is disrupted by the insertion of random frames, which may distract the user [70]. However, human neural mechanisms, coupled with the properties of IM make the performance of human users far superior to automated computer recognition programmes.

5.5 Conclusions

In this chapter, the second proposed CAPTCHA model, VICAP v.2, was introduced, which is an improved version of the first CAPTCHA model, VICAP v.1. As has been discussed, the security weakness that was making VICAP v.1 vulnerable to image-processing attacks was rectified in the design of VICAP v.2. The proposed second CAPTCHA model has been tested on both human users and the most advanced CAPTCHA decoders using our state-of-the-art CTA, and the results of these tests have been analysed.

Over 700 experiments have been conducted on both human users and computer-based attacks, and we have gained an appropriate level of knowledge in terms of security and usability of VICAP v.2. From our research, we discovered that character recognition success rates for human users compared to computer-based recognition programmes would ideally increase to 100% when the ORO parameter is set at 40%. This will generate an HRSR of 100% and a CRSR of 50% for the entire CAPTCHA string, as well as keep the CRSR to almost 0% per individual frame.

In addition to that, the proposed model was tested against different attacks and the quantitative output results were analysed using different recognition techniques. As a result of conducting a variety of experiments using different inputs, it can be concluded that, with an OSR of 25% and a background noise of 15%, having a high ORO value will make the test easier for attackers to break. It has also been identified that by decreasing the value of the ORO parameter to around 30%, the CRSR will also dramatically decrease. Thus, the CRSR will continue to drop until it reaches 20%, which is equal to 0%. However, in this scenario, it would still be possible for human users to recognise the CAPTCHA characters when the ORO parameter is equal to 20%, with a 65% accuracy level.

Chapter 6

Conclusion and Future Work

This chapter aims to provide a conclusion to all the information presented in this thesis, and also provide some information about the future of CAPTCHAs and online authentication methods.

6.1 Conclusion

In essence, the aim of this thesis was to demonstrate the importance of online security, especially in the field of e-commerce and social networking, where there is a large amount of personal information involved. The number of internet users in the world is growing rapidly and many people rely on the internet to meet their personal, financial, and professional needs. Therefore, there are growing concerns about the security of online services and companies are under pressure to provide safer and more reliable online environments for their customers. This thesis has discussed a number of system authentication methods that are often used and also defined what is meant by a CAPTCHA. Additionally, several types of CAPTCHA, including text-based and non-text-based models were compared and analysed in order to present the advantages and disadvantages of each. By providing a comprehensive analysis of the state-of-the-art in the field of CAPTCHAs, the idea of a CAPTCHA model that solely relies on a uniquely human ability that cannot be imitated by computers was presented. This model adheres to the definition of a CAPTCHA, which is clearly defined as a test that humans can pass, but computer programmes cannot. CAPTCHAs can be integrated into any online form using a simple scripting language and CAPTCHA engines are usually placed in a location where a client application can communicate with the server by setting up a secure communication channel. Depending on whether the requested resources are protected or not, the web server application can obtain help from the CAPTCHA server to automatically decide whether the user is a human or an automated bot. CAPTCHAs can be used in many different online applications, such as voting, ticket sales, chatrooms, and many more. The point of using a CAPTCHA is to automatically filter out automated attackers with minimal time and effort.

As discussed previously, CAPTCHAs can be categorised according to their specifications and the way they have been designed. One of the most popular types of CAPTCHAs is called text-based or OCR-based CAPTCHAs. They are called text-based CAPTCHAs because they use text or characters, which are then distorted and presented to the user. The user is then required to recognise and decipher these characters. These types of CAPTCHAs are the most popular because they are easy to make and not costly. However, there are a lot more threats and attacks associated with these types of CAPTCHA. Another CAPTCHA type is called a non-OCR-based CAPTCHA, which is made up of purely non-text character material, such as images of natural scenes or specific objects. These types of CAPTCHAs are more costly to generate and more time consuming for the user to solve. For this reason, non-OCR-based CAPTCHAs or image-based CAPTCHAs are not as popular as text-based CAPTCHAs. Non-OCR-based CAPTCHAs require sophisticated image recognition software to be hacked. On the other hand, they are also sometimes very difficult even for human users to understand and solve. Another type of CAPTCHA is called non-visual based CAPTCHAs, which use short audio and voice clips. These types of CAPTCHAs are in the minority because they are known to be the most difficult and costly to make and most difficult for the end user to solve. As the results of this PhD research confirm, text-based CAPTCHAs are the most suitable and popular type of CAPTCHA for human users and most people prefer this type to other types. For this reason, in this research project, I have worked on a new text-based CAPTCHA model called VICAP, which is based on the unique ability of humans to superimpose and integrate a set of fleeting frames using their visual short-term memory.

Since different categories of CAPTCHA challenges have been discussed, the different types of attacks and threats associated with each type were also outlined. OCR attacks are known to be the most significant type of attack to OCR-based CAPTCHAs. OCR software uses distinctive character recognition algorithms to break CAPTCHA challenges. The hacking or breaking of any text-based CAPTCHA requires different intricate processing stages. OCR programmes are able to recognise and break any current text-based CAPTCHA by segmenting and separating the characters. Character segmentation is a key character recognition technique. Since our proposed CAPTCHA model presents only partial information on a single frame scenario, theoretically it would not be possible for any current OCR programme to distinguish or separate individual characters or numbers in the CAPTCHA string. Thus, our proposed model would be highly robust against current OCR technologies.

Third party human attacks are known to be the second most common and dangerous type of attack. This method uses cheap labour across different parts of the globe in order to seize and send thousands of CAPTCHA images to third-party human solvers, who solve the CAPTCHAs in seconds and send the answers back to the online forms they came from. This is known as one of the cheapest and most accurate types of CAPTCHA attack because these CAPTCHAs are designed to be recognised and distinguished by humans. Therefore, humans can solve and recognise most current CAPTCHA models more easily and efficiently than current OCR programmes in terms of cost. Our proposed CAPTCHA model uses a series of frames that the user is required to watch in order to obtain the correct CAPTCHA characters using what is known as Persistence of Vision. Therefore, even by taking a screen-shot, no useful information will be retrieved by groups of human hackers in a single frame scenario. This thesis has presented a comprehensive comparison of different types of CAPTCHA attacks, and the advantages and disadvantages of each have been investigated. Several types of OCR-based and non-OCR-based CAPTCHAs have also been compared in terms of usability and security, and the output results of these have been presented and compared. As the analytical output results confirm, OCR-based CAPTCHAs are known to be the most popular type of CAPTCHA. Text-based CAPTCHAs are easy to set up and very cost-efficient compared to non-OCR or non-visual based CAPTCHAs, which are more expensive to set up and take longer to make and render. They also require less time and energy on the part of the user to solve compared to other non-OCR or non-visual CAPTCHA types. In terms of security, OCR-based-CAPTCHAs are rated as less secure than non-OCR and non-visual CAPTCHA types, which have been rated as average-high in terms of their robustness against different attacks and recognition algorithms.

As part of this PhD research, we have analysed different elements and issues that can affect the level of usability and performance of the CAPTCHA test. These elements can be summarized in three categories: Distortion, Content, and Presentation. All of these elements can have a very significant impact on the usability and performance of different CAPTCHA models. For example, having too much distortion on an image might cause the image to become so blurred that a user is unable to read it. Using different colours for the background or foreground of a CAPTCHA can sometimes cause part of the characters to be hidden, and that again may cause the characters to be become over ambiguous for a user to read. After identifying these security and usability issues, it is now prudent to experiment with real-world examples.

As mentioned at the end of Chapter 2, different examples of popular CAPTCHA models have been analysed in order to discover how they can be broken. As it can be seen from the results presented in Chapter 2, it is vital to creating a new CAPTCHA model that can be resistant to current attacks, especially OCR programmes. Our CAPTCHA model is a test that only human users, not computer bots, should be able to understand. A robust CAPTCHA model can be defined as a model that only relies on unique human abilities. POV is known as one of the unique abilities of the human eye, and therefore, computer recognition software would not be able to imitate this ability. POV is the main reason that we can see the world around us in a uniform and integrated manner by superimposing and integrating all the images our eyes see using a sophisticated form of memory called Iconic Memory. IM enables images to be placed inside our short-term memory for a fraction of a second. There are key differences between short-term memory and IM. A major difference is the amount of time that visual information can be held in our visual system before being wiped from our IM. IM enables the human brain to superimpose and integrate a series of fleeting frames into a final image, which cannot be achieved with short-term memory. POV is very important to our research because it has enabled our new CAPTCHA model to be robust because it is only understandable to humans.

Psychophysics is the study of the relationship between a human brain and its surrounding environment, and how this information is translated using its sophisticated neural network. Essentially, every signal stimulus placed in front of the human eye will remain in our visual system for about a tenth to a fifteenth of a second before disappearing. Therefore, to create POV, it is important that a new image appears before the effect of the previous image has disappeared. For example, in the case of our new CAPTCHA model, if two frames (or stimuli) appear within the same time window, the result will be the superimposition of two images. This is the idea of our proposed VICAP model. The frame per second (FPS) rate is an essential variant to enable POV because a slow frame rate (of less than 16 FPS) will cause our eyes to see flashing images only, rather than a smooth-running series of frames. This is the main reason that current movie technologies work at a rate of 24 FPS, a rate that will ensure that our visual system sees the series of frames as one smooth running film. Trans-saccadic movement is the smallest movement of the human eye, which will cause the HVS to integrate perceived visual information. As has been discussed in this thesis, two scenarios have been proposed in order to explain the procedure of integrating visual information using trans-saccadic memory. The Single Stage Scenario was introduced in order to build and render every single frame of our proposed CAPTCHA model based on the correlation between the OSR and background noise. In order to achieve the effect of POV, these trans-saccadic moments need to be repeated quickly, which will cause visual integration. This is called the Multi-Stage Scenario, by which we have proposed a mathematical model of the trans-saccadic integration technique.

Our proposed VICAP model consists of a range of different stages and phases. A state-of-the-art application has been developed as part of this research entitled the CAPTCHA-Generator Application. The role of the CGA is to render and generate single VICAP frames according to the defined specifications and then play those frames to the user at high speed. The rendering stage will commence by sampling the object and then cause a binary conversion. Binarisation will make the object easier for the human eye to recognise. This is due to the number of photoreceptors that exist in a healthy human eye. Every human eye is made up of two sets of photoreceptors, rods and cones. Rods are sensitive to different shades of grey (and also black and white), whilst cones are sensitive to colour. However, there are many more rods than cones in the human eye, and so our eyes react more efficiently to black and white images, rather than colourful ones.

In order to test our proposed CAPTCHA model, another state-of-the-art application was developed called CAPTCHA-Test Application. The CTA works in a very similar way to the human eye. This application is able to superimpose a series of frames by calculating the total number of pixels in a single frame scenario and after comparing this information, make a graphical representation of the final output results. This final graphical output representation would be very similar to the results that a human eye would see. Yet, as has been discussed, this would be the only disadvantage of our proposed CAPTCHA model (version.1) as it caused security concerns.

In order to overcome these security concerns, the second CAPTCHA model was proposed called VICAP (version. 2), which is the improved version of the first VICAP model. The frame sequence of VICAP v.2 benefitted from being injected with random frames alongside original frames to disturb the total number of appearing pixels in each sequence. By modifying the CGA, it would be able to generate and render two lines of frames concurrently. Production Line-1 will generate the original VICAP frames with a specific OSR and background noise level, and Production Line-2 will generate random frames only. These two generated frame sequences will be mixed together using a parameter called Original to Random Output (ORO) data. ORO parameters play a key role in our second proposed CAPTCHA model because the higher the value of the ORO, the higher the number of original frames compared to random frames. However, this would make the CAPTCHA easier for both human users and computer recognition software to recognise. Alternatively, the lower the value of the ORO, the lower the number of original frames compared to random frames in the frame sequence. However, in this instance, the human eye would not be able to pick up sufficient information about the object and so would not be able to recognise it. What we can understand from this is that there is a trade-off between the ratio of the ORO parameter and the recognition success rates. This trade-off is between the usability and the security of the CAPTCHA. A higher ORO parameter will increase the usability of the CAPTCHA model but will decrease its security level and vice versa. In order to find the optimal value for the ORO parameter, a comprehensive series of experiments were conducted in laboratory conditions that tested a variety of object sampling and background noise level rates. The VICAP frames were generated using the CGA and the output rendered images were superimposed and integrated using the CTA. As a result of these experiments, the optimal level of the ORO parameter was measured at 40%, which gives a CRSR of 50% and an HRSR of 50% based on a multi-frame scenario. Additionally, the CRSR will still be maintained at 0% per individual frame. In total, over 400 experiments were conducted to test a range of object sampling and BNRs against the CRSR.

CAPTCHA User Experience and Performance (CUEP) website were designed in order to gain valuable user feedback from a various group of participants. Over 150 participants were involved in this research evaluation programme and recorded their thoughts on various aspects of the usability and performance of two CAPTCHAs, including our VICAP v.2 model. A comprehensive user questionnaire that included 13 different questions was designed and presented to the participants in order to measure different usability elements qualitatively and quantitatively. Our proposed CAPTCHA model was compared with one of the most famous current Google CAPTCHA models called ReCAPTCHA. The output results confirm that our proposed VICAP model is superior in terms of usability compared to ReCAPTCHA. For example, in terms correct response attempt rates, 98% of participants solved VICAP v.2 on their first attempt compared 61%, who solved the ReCAPTCHA on their first attempt. Additionally, 92% of participants solved our VICAP v.2 in less than 3 seconds, while only 10% of participants solved ReCAPTCHA in less than 3 seconds. Lastly, 36% of participants reported that they would be 'very likely' to participate in solving the VICAP v.2 again in future, as opposed to only 4% for the ReCAPTCHA. Another important element that could affect the usability and performance of the CAPTCHA test is the difficulty level of the challenge. However, 65% of participants labelled the VICAP v.2 model as 'very easy' to solve, as opposed to 4% of participants for the ReCAPTCHA. The clarity level of the characters is also an important aspect of CAPTCHA usability. 35% of participants labelled the VICAP v.2 characters as 'very clear,' as opposed to 0% of participants for the ReCAPTCHA. 56% of participants labelled the VICAP v.2 model as 'moderately clear,' as opposed to only 8% for the ReCAPTCHA. Lastly, the size of the characters can also affect the usability and performance of a CAPTCHA challenge. 80% of participants labelled the character size of the VICAP v.2 model as being 'just right,' as opposed to only 41% for the ReCAPTCHA. These user experience results clearly prove that our proposed VICAP v.2 is better in terms of usability and performance compared to other popular CAPTCHA models.

6.2 Future Direction of CAPTCHA Research

As has been discussed in this thesis, one of the main challenges in the field of cybersecurity and online authentication systems is to distinguish between real human users and computer robots. Online service providers such as online shops, online banking services, email providers, and social networks such as Facebook and Twitter must offer robust security features for their customers that can resist automated computer attacks. The reliance of the public on online services only highlights how important it is to invest and conduct further research into the field of internet security and specifically, CAPTCHAs. All CAPTCHA models that are currently being used across the web utilise only two human senses: vision and hearing. Text-based and image-based CAPTCHAs are very popular and these ugly distorted images can be seen almost everywhere, including websites like Facebook, Yahoo, and PayPal. Audible CAPTCHAs are also being used to accommodate those who are visually impaired. Yet, the majority of CAPTCHA models are very time consuming and users often spend too much time and energy solving them. As our proposed VICAP v.2 is a model based on POV, solving time has been cut to a minimal. CAPTCHA research will continue to concentrate on new types of CAPTCHAs that not only improve the security of websites by avoiding bots and spams attacks but also improve the quality of authentication methods.

Bibliography:

[1] Symantec Corporation, “Crimeware: Bots,” Symantec Corporation, 2014. [Online]. Available: http://uk.norton.com/cybercrime-bots. [Accessed July 2014].

[2] Monica Chewa and Henry S. Baird, “BaffleText: a Human Interactive Proof,” in *SPIE/IS&T Document Recognition & Retrieval*, Santa Clara, CA, January 2003.

[3] Kumar Chellapilla, Kevin Larson, Patrice Simard and Mary Czerwinski, “Designing Human-Friendly Human Interaction Proofs (HIPs),” in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, Portland, Oregon, USA, 2005.

[4] Carnegie Mellon University, “CAPTCHA: Telling Humans and Computers Apart Automatically,” Carnegie Mellon University, 2010. [Online]. Available: http://www.captcha.net/. [Accessed July 2015].

[5] Luis von Ahn, Manuel Blum, John Langford, “Telling Humans and Computers Apart Automatically.,” *Communications Of The ACM*, vol. 47, pp. 57-60, February 2004.

[6] Luis von Ahn, Benjamin Maurer, Colin McMillen, David Abraham and Manuel Blum, “reCAPTCHA: Human-Based Character Recognition via Web Security Measures,” *www.sciencemag.org*, vol. VOL 321, pp. 1465-1468, 12 September 2008.

[7] Turing, A. M. “Computing Machinery and Intelligence.” Mind, vol. 59, no. 236, 1950, pp. 433–460. JSTOR, JSTOR, www.jstor.org/stable/2251299.

[8] Daniel Bates, “Mail Online,” Associated Newspapers Ltd, 25 May 2011. [Online]. Available: http://www.dailymail.co.uk/sciencetech/article-1390796/Captcha-cracked-Security-fears-online-word-test-proved-vulnerable-hacking.html. [Accessed 20 October 2017].

[9] Ayse Pinar SayginIlyas CicekliVarol Akman, “Turing Test: 50 Years Later,” *Minds and Machines,* no. 10, p. 463–518, 2000.

[10] CharlesGillingham, “Weakness of Turing test,” [Online]. Available: http://commons.wikimedia.org/wiki/File:Weakness_of_Turing_test_1.svg#mediaviewer/File:Weakness_of_Turing_test_1.svg. [Accessed Jan 2014].

[11] Giacomo Parmeggiani , “Accessibility of CAPTCHAs,” BeSpecular (Pty) LTD, 27 Jan 2016. [Online]. Available: http://www.bespecular.com/blog/accessibility-of-captchas/. [Accessed October 2017].

[12] Luis von Ahn, Manuel Blum, Nicholas J. Hopper, John Langford, “CAPTCHA: Using Hard AI Problems For Security,” 2000.

[13] Luis von Ahn, “Telling Humans and Computers Apart Automatically,” Carnegie Mellon University, 2000-2010. [Online]. Available: www.captcha.net.

[14] C. Johnson, “Vicarious AI Passes Turing Test,” 28 October 2013. [Online]. Available: https://www.eetimes.com/document.asp?doc_id=1319914. [Accessed Oct 2017].

[15] Moradi, M. and Keyvanpour, M., “CAPTCHA and its Alternatives: A Review,” *Security and Communication Networks,* vol. 8, no. 12, 2015.

[16] Shirali-Shahreza, M. and Shirali-Shahreza, S., “Collage CAPTCHA,” in *Signal Processing and Its Applications, 2007. ISSPA 2007. 9th International Symposium on,* 2007, pp. 1-4.

[17] Luis von Ahn, Manuel Blum and John Langford, “Telling Humans and Computers Apart Automatically.,” *Communications Of The ACM,* Vols. 47, No. 2, pp. 57-60, February 2004.

[18] J. Elson, J. Douceur, J. Howell and J. Saul, “Asirra: a captcha that exploits interest-aligned manual image categorization,” in *In ACM Conference on Computer and Communications Security,* 2007, p. 366–374.

[19] Te-En Wei, Jeng, A.B. and Hahn-Ming Lee, “GeoCAPTCHA - A novel personalized CAPTCHA using the geographic concept to defend against 3rd Party Human Attack,”, *2012 IEEE 31st International Performance Computing and Communications Conference (IPCCC),* Austin, TX, 2012, pp. 392-399.

[20] Haichang Gao, Dan Yao, Honggang Liu, Xiyang Liu and Liming Wang, “A Novel Image-Based CAPTCHA Using Jigsaw Puzzle,” in *Computational Science and Engineering (CSE), 2010 IEEE 13th International Conference on*, 2010, pp. 351-356.

[21] Jeff Yan, Ahmad Salah El Ahmad, “A Low-cost Attack on a Microsoft CAPTCHA,” Newcastle University, UK, 2008.

[22] Vu Duc Nguyen, Yang-WaiChow and WillySusilo, “On the security of text-based 3D CAPTCHAs,” *Computers & Security,* vol. 45, pp. 84-99, September 2014.

[23] C.Woodford, “Optical character recognition (OCR),” 20 November 2013. [Online]. Available: http://www.explainthatstuff.com/how-ocr-works.html. [Accessed 24 September 2013].

[24] Nicomsoft, “Optical Character Recognition (OCR) – How it works,” Nicomsoft Ltd, 5 February 2012. [Online]. Available: http://www.nicomsoft.com/optical-character-recognition-ocr-how-it-works/. [Accessed 24 September 2014].

[25] Jeff Yan and Ahmad Salah El Ahmad, “A Low-cost Attack on a Microsoft CAPTCHA,” 2008. [Online]. Available: http://www.lancaster.ac.uk/staff/yanj2/msn_draft.pdf

[26] Jeff Yan and Ahmad Salah El Ahmad, “CAPTCHA Security: A Case Study,” *IEEE Security & Privacy*, vol. 7, pp. 22-28, 2009.

[27] A.A.Chandavale and A. M. Sapkal, “Algorithm for Secured Online Authentication Using CAPTCHA,” in *3rd International Conference on Emerging Trends in Engineering and Technology,* Goa, 2010.

[28] Greg Mori and Jitendra Malik, “Breaking a Visual CAPTCHA,” 2002. [Online]. Available: http://www.cs.sfu.ca/~mori/research/gimpy/#approach.

[29] Jeff Yan and Ahmad Salah El Ahmad, “Breaking Visual CAPTCHAs with Naive Pattern Recognition Algorithms,” in *Twenty-Third Annual Computer Security Applications Conference (ACSAC 2007),* Miami Beach, FL, 2007, pp. 279-291.

[30] J.Yan and A. S. El Ahmad, "CAPTCHA Security: A Case Study," in IEEE Security & Privacy, vol. 7, no. 4, pp. 22-28, July-Aug. 2009.

[31] Yannis Soupionis and Dimitris Gritzalis, “Audio CAPTCHA: Existing solutions assessment and a new implementation for VoIP telephony” *computers & security,* pp. 603-618, 2010.

[32] Moy, G., Jones, N., Harkless, C. and Potter, R., “Distortion estimation techniques in solving visual CAPTCHAs,” *IEEE Computer Society Conference on Computer Vision and Pattern Recognition,* vol. 2, no. 1063-6919, pp. II-23-II-28 Vol.2, 2004.

[33] Truong, H.D., Turner, C.F. and Zou, C.C., “iCAPTCHA: The Next Generation of CAPTCHA Designed to Defend against 3rd Party Human Attacks,” *Communications (ICC), 2011 IEEE International Conference on,* pp. 1-6, 2011.

[34] B. Acohido, “Cybergangs use cheap labour to break codes on social sites,” USA TODAY, 23 April 2009. [Online]. Available: http://usatoday30.usatoday.com/tech/news/computersecurity/2009-04-22-captcha-code-breakers_N.htm. [Accessed 07 March 2013].

[35] D. Danchev, “Inside India's CAPTCHA solving the economy,” Zero Day, 29 August 2008. [Online]. Available: http://www.zdnet.com/blog/security/inside-indias-captcha-solving-economy/1835. [Accessed 07 March 2013].

[36] Luis von Ahn, Benjamin Maurer, Colin McMillen, David Abraham and Manuel Blum, “reCAPTCHA: Human-Based Character Recognition via Web Security Measures,” *www.sciencemag.org,* vol. VOL 321, pp. 1465-1468, 12 September 2008.

[37] Jakob Nielsen, “Usability 101: Introduction to Usability,” Nielsen Norman Group, 4 January 2012. [Online]. Available: http://www.nngroup.com/articles/usability-101-introduction-to-usability/. [Accessed 03 June 2015].

[38] Jeff Yan and Ahmad Salah El Ahmad, “Usability of CAPTCHAs Or usability issues in CAPTCHA design,” in *Symposium On Usable Privacy and Security (SOUPS),* Pittsburgh, PA, USA, 2008, July 23-25.

[39] Marc, “CAPTCHA The Moment,” 26 March 2011. [Online]. Available: http://mkcohen.com/2011/03. [Accessed June 2015].

[40] C. George Boeree, “Gestalt Psychology,” 2000. [Online]. Available: http://webspace.ship.edu/cgboer/gestalt.html. [Accessed June 2015].

[41] Chandavale, A.A. and Sapkal, A.M, "Algorithm for Secured Online Authentication Using CAPTCHA," in *Emerging Trends in Engineering and Technology (ICETET), 2010 3rd International Conference on*, 2010, pp. 292-297.

[42] Jeff Yan and Ahmad Salah El Ahmad, "A Low-cost Attack on a Microsoft CAPTCHA," in *CCS '08 Proceedings of the 15th ACM conference on Computer and communications security*, ACM New York, NY, USA, 2008.

[43] Monica Chew and Henry S. Baird, "BaffleText: a Human Interactive Proof," in *SPIE/IS&T Document Recognition & Retrieval*, Santa Clara, CA, 2003.

[44] Allison L. Coates, Henry S. Baird and Richard J. Fateman, "Pessimal Print: A Reverse Turing Test," in *Proceedings of Sixth International Conference on Document Analysis and Recognition*, Seattle, WA, 2001, pp. 1154-1158.

[45] Ahmad El Ahmad, Jeff Yan and Wai-Yin Ng, "CAPTCHA Design Color, Usability, and Security," in *IEEE Internet Computing,* vol. 16, no. 2, pp. 44-51, March-April 2012.

[46] Abelard, "The Theory of Colour," December 2013. [Online]. Available: http://www.abelard.org/colour/col-hi.htm. [Accessed June 2015].

[47] "Rods and Cones," [Online]. Available: http://hyperphysics.phy-astr.gsu.edu/hbase/vision/rodcone.html. [Accessed June 2015].

[48] Von Ahn, Luis, Blum, Manuel and Langford, John, "Telling Humans and Computers Apart Automatically.," *Communications Of The ACM,* vol. 47, pp. 57-60, February 2004.

[49] Seyed Mohammad Reza Saadat Beheshti and P. Liatsis, "VICAP: Using the mechanisms of trans-saccadic memory to distinguish between humans and machines," London, 2015, pp. 295-298.

[50] Chris Clause, "Iconic Memory: Definition, Examples & Quiz," Education Portal, 2003-2014. [Online]. Available: http://education-portal.com/academy/lesson/iconic-memory-definition-examples-quiz.html#lesson. [Accessed October 2014].

[51] M. McKinney, "The Persistence of Vision," Spring 2008. [Online]. Available: http://www.vision.org/visionmedia/article.aspx?id=136. [Accessed October 2014].

[52] David E. Irwin and Laura E. Thomas, “Visual Sensory Memory,” in *Visual Memory*, Oxford University Press, 2008.

[53] Steven J. Luck and Andrew Hollingworth, Visual Memory, Oxford Scholarship Online, September 2008.

[54] Final Cut Pro 7, “How Many Frames per Second Is Best?,” Apple, [Online]. Available: https://documentation.apple.com/en/finalcutpro/usermanual/index.html#chapter=D%26section=3%26tasks=true. [Accessed Oct 2014].

[55] M. Armstrong, D. Flynn, M. Hammond, S. Jolly and R. Salmon, “High Frame-Rate Television,” in *BRITISH BROADCASTING CORPORATION*, September 2008.

[56] G. McConnell, “Gordon McConnell at Broschofsky Galleries,” Broschofsky Galleries, 2014. [Online]. Available: http://www.brogallery.com/artists-links/gordon-mcconnell/. [Accessed 2014].

[57] K. Rayner, “Eye Movements in Reading and Information Processing: 20 Years of Research,” *Psychological Bulletin*, vol. 124, pp. 372-422, 1998.

[58] David E. Irwin, “Integrating Information across Saccadic Eye Movements,” *Current Directions in Psychological Science*, vol. 5, no. 3, pp. 94-100, June 1, 1996.

[59] David Irwin, “Information Integration across Saccadic Eye Movements,” *Cognitive Psychology*, vol. 23, pp. 420-456, 1991.

[60] Seyed Mohammad Reza Saadat Beheshti and P. Liatsis, “CAPTCHA Usability and Performance; How to Measure the Usability Level of Human Interactive Applications Quantitatively and Qualitatively?,” in *Developments of eSystems Engineering (DeSE)*, Dubai, 2015, pp. 131-136.

[61] Understand graphics formats, “Adobe”, December 2015. [Online]. Available: http://help.adobe.com/en_US/indesign/cs/using/WSa285fff53dea4f8617383751001ea8cb3f-6be2a.html#WSa285fff53dea4f8617383751001ea8cb3f-6be1a.

[62] HyperPhysics, “HyperPhysics,” [Online]. Available: http://hyperphysics.phy-astr.gsu.edu/hbase/vision/rodcone.html#c3. [Accessed June 2016].

[63] Software development and Analytics, “GSA Captcha Breaker,” 2014. [Online]. Available: http://captcha-breaker.gsa-online.de/. [Accessed December 2015].

[64] Captcha Sniper, “Captcha Sniper,” 2016. [Online]. Available: http://www.captchasniper.com/new/index.html. [Accessed July 2016].

[65] GSA Captcha Braker, “GSA Software development and Analytics,” 2016. [Online]. Available: https://captcha-breaker.gsa-online.de/. [Accessed July 2016].

[66] Luis von Ahn, Manuel Blum, Nicholas Hopper and John Langford, “CAPTCHA: Telling Humans and Computers Apart Automatically,” Carnegie Mellon University, 2000-2010. [Online]. Available: http://www.captcha.net/. [Accessed June 2015].

[67] Seyed Mohammad Reza Saadat Beheshti, “Captcha User Experience Web Page,” City University London, July 2015. [Online]. Available: https://fs22.formsite.com/Mrbeheshti/form4/index.html. [Accessed August 2015].

[68] Seyed Mohammad Reza Saadat Beheshti, “CAPTCHA Usability and Performance Programme,” City University London, July 2015. [Online]. Available: http://mrbeheshti2.wixsite.com/captcha-project. [Accessed August 2015].

[69] Charan Ranganath and Gregor Rainer, “Neural mechanisms for detecting and remembering novel events,” *Cognitive Neuroscience,* vol. 4, pp. 193-202, 2003.

[70] N. Kumar, Kiran, Chandramouli, Suyog and Shiffrin, Richard, “Salience, perceptual dimensions, and the diversion of attention,” *The American Journal of Psychology,* vol. 128, no. 2, pp. 253-265, 2015.

[71] Seyed Mohammad Reza Saadat Beheshti, “Online Banking,” City, University of London, 2016. [Online]. Available: http://mrbeheshti2.wixsite.com/captcha-project/online-banking. [Accessed 2017].

[72] Lindsay W. MacDonald, “Using Color Effectively in Computer Graphics,” in *IEEE Computer Graphics and Applications*, University of Derby, UK, July/August 1999.

[73] Naumann A.B., Franke T., Bauckhage C, “Investigating CAPTCHAs Based on Visual Phenomena,” in *IFIP International Federation for Information Processing*, 2009. vol 5727. Springer, Berlin, Heidelberg

[74] J. Lung, “Ethical and legal considerations of reCAPTCHA,” *2012 Tenth Annual International Conference on Privacy, Security and Trust*, pp. 211-216, 2012.

[75] Greg Mori and Jitendra Malik, “Recognizing Objects in Adversarial Clutter: Breaking a Visual CAPTCHA,” 2003. University of California, Berkeley, CA 94720.

[76] S. Robinson, “Can Hard AI Problems Foil Internet Interlopers?,” The CAPTCHA Project, 2001. [Online]. Available: http://www.captcha.net/news/ai.html. [Accessed 29 01 2013].