

# Devanagari CAPTCHA: For the Security in Web

<sup>[1]</sup> Mohinder Kumar, <sup>[2]</sup> Sanjeev Kumar

<sup>[1]</sup> Department of Computer Science & Applications, Panjab University Regional Centre, Muktsar,  
PB INDIA

<sup>[2]</sup> Department of Computer Science, D.A.V. College, Abohar, PB INDIA

E-mail: <sup>[1]</sup> kumarmohinder@pu.ac.in, <sup>[2]</sup> gumber\_sanjeev@yahoo.com.

**Abstract:** Completely Automated Public Turing Test to Tell Computers and Humans Apart or CAPTCHA is a solution for cyber-attack. CAPTCHA is a small challenge that an internet user has to pass before accessing any online service. The most common type of CAPTCHA is text-based CAPTCHA, in which a small image (contains a random number of alphabets) is presented before the user. The user has to identify and then type the alphabet in a text box. The textual information in the CAPTCHA must not be identified by a bot (computer code). So, artificial noise and distortion are applied in the image. Earlier text-based schemes use English alphabets, but over time non-English language-based text CAPTCHAs also came into the picture. Native language-based text CAPTCHA is very useful for internet users who do not know the English language. This article is an effort towards the current status of the Devanagari script-based CAPTCHAs. We have analyzed 28 unique Devanagari CAPTCHAs from a security and usability point of view. Total 28000 different samples are collected for this experiment. For the success of a text-based CAPTCHA, it must be very secure from the bot and easy for human beings. Devanagari CAPTCHA can be very beneficial for Indian websites. This paper is written by keeping the importance of Devanagari script-based CAPTCHA.

**Keywords:** Devanagari CAPTCHA, Hindi CAPTCHA, Cyber Security, Bot, Reverse Turing Test, De-CAPTCHA

## 1. Introduction

We are living in the world of internet services. These online services have built our life very secure as well as comfortable. Most of our routine activities are being done through online mode during the current pandemic time. The organizations are converting their services according to this online world. What happens if suddenly we come to know that these services are also vulnerable to attack? Such an attack on the websites is known as a cyber-attack. Cyber-attack is one of the most dangerous attacks in today's world. These attacks have a bigger impact, and the losses are very high. The owners of the websites spend a heavy amount of money for a high level of security so that the clients get a safer environment. These solutions are also very costly. This cost puts a financial burden both on the owners and customers of the organizations. In this attack, computers are trained to breach the security of the websites. A website that is under the control of a computer program can be compromised in many ways. The research community is always busy finding a better alternative for cyber security. Completely Automated Public Turing Test to Tell Computers and Humans Apart or CAPTCHA, in short, is such an alternative to this attack. It is a cost-effective solution and also very easy to implement. CAPTCHA needs to be designed very intelligently so that only human users can use it. Humans always find it very enjoyable to pass a small challenge in the form of puzzles. This hypothesis is the origin of CAPTCHA. Now the websites give a small challenge appearing as CAPTCHA to the users that have to pass before accessing the websites. The challenge must be easy for humans to solve while very difficult to solve by a computer program. It is hard to design such a challenge at this time that is not solved by the machines. It is the reason that different types of CAPTCHA are proposed over time. These types include text-based, image-based, audio-based, puzzle-based, and mouse-based. For the success of a CAPTCHA, it must possess both criteria: usability and security. In this paper, we have been concerned only with text-based schemes in general and Devanagari script-based text CAPTCHA in particular. Devanagari CAPTCHA is never tested from the security point of view. It is also true that Devanagari CAPTCHA is not used for the websites, but it may be used in future. The popularity of native languages in the websites is creating a platform for native language-based CAPTCHA

schemes. The field of OCR is also not as mature for the native languages as in the case of English language. This fact can be exploited by using Devanagari CAPTCHA as a security mechanism on the websites. This paper highlights the issues related to the security and usability of the Devanagari CAPTCHA. For Devanagari CAPTCHA, it is the first effort till this time that includes both aspects. This paper consists of seven sections. In section 2, the related work has been discussed. In section 3, the motivation behind this work is justified. In section 4, the corpus and the generation techniques for Devanagari CAPTCHAs are discussed. Section 5 is devoted to the analysis of the collected samples. Section 6 highlights the future scope of this work, and the final section concludes the work.

## 2. Related work

Text-based CAPTCHA exists majorly in the English language. Gimpy CAPTCHAs are the first text-based CAPTCHA that came into the picture in 2003 [1]. Gimpy CAPTCHA was a series of CAPTCHA in which monochrome, colored easy and complex schemes were present. Baffle CAPTCHA, Pessimist Print CAPTCHA [2], Clickable CAPTCHA [3], MSN CAPTCHA [4], Sigma-Lognormal CAPTCHA [5], Crowding Character Together CAPTCHA [6], 3-D CAPTCHA [7], STE3D-CAP [8], and Mixed Text Synthetic Handwritten CAPTCHA [9] are some of the popular English language-based text CAPTCHA schemes. Literature shows that these English language-based CAPTCHAs are successfully compromised. In 2003, the breaking of Gimpy CAPTCHAs with shape and context matching techniques is reported with a success rate of 33% to 92% [10]. Gimpy CAPTCHA is again broken in 2004 with a success rate of 99% breaking rate by using the concept of distortion estimation methods [11]. In the same year, six popular CAPTCHAs like Google, Yahoo, and Mailblocks are broken with simple segmentation and recognition techniques [12]. In 2012, several color schemes like Gimpy-r, EZ-Gimpy, LinkedIn, FreeCap, BotBlock, Megaupload, BotDetect, and phpCAPTCHA.org were broken successfully with the color filling segmentation method [13]. In 2013, some hollow CAPTCHAs were broken with the color filling segmentation method [14]. In 2014, an algorithm was proposed to break 3D CAPTCHA like Super CAPTCHA, 3dCAPTCHA, and TeaBag3D 1.2 [15]. In 2016, a genetic algorithm was proposed to break every text-based scheme. It uses a Log-Gabor 2D filter to extract the characters from the CAPTCHA without preprocessing [16]. In 2019, a Deep Learning-based attack is reported to break Roman CAPTCHA and Chinese CAPTCHA [17]. Very few non-English language-based CAPTCHAs exist, e.g., Handwritten Arabic CAPTCHA [18], Chinese CAPTCHA [19], Gurumukhi CAPTCHA [20], and Deva CAPTCHA [21, 22]. It is proven that native language-based CAPTCHA is more usable for the native people of the region [19]. That is the reason for the existence of non-English-based CAPTCHAs. Being the native citizen of India, this motivated us to present a study of the existing Devanagari language-based CAPTCHAs.

## 3. Motivation


















Hindi is the most spoken language of India that is written in Devanagari script. Local language-based CAPTCHAs are more usable for local citizens of India as there is no language barrier while using the websites. Many of the websites, especially government websites, are accessed by the majority of the population. Such websites represent data in their native language for the sake of understanding by each citizen. To provide security from the bot attack, CAPTCHA is also used on these native websites. The contents of a CAPTCHA must be based on native language. The people ratio that can get the benefits of the public sector will decrease otherwise. Some Devanagari script-based CAPTCHAs are reported in the literature, but no website has implemented these schemes. A popular CAPTCHA web development organization is BotDetect which is selling approximately 60 types of CAPTCHA designs. These CAPTCHAs are based on different scripts that cover almost every language in the world. BotDetect company is selling these designs in the market to government and private organizations in which NASA, United States Department, Dell, Hewlett Packard, HDFC Ltd. Australia Drug Foundations are included. This company covers the continents that include North America, Europe, Asia, Australia, New Zealand, South America, Africa, and Antarctica. The cost of these companies is very high. The designs look very typical for a CAPTCHA scheme. Devanagari CAPTCHA schemes attracted us because currently these are not being used by the Indian websites, but may be used in the future. The government of India has already taken the initiative to replace the contents of Indian websites with native languages. It is done for the native citizens so that they can be benefitted from the public schemes. It is also









possible to use the security measures in the native language. CAPTCHA is the most commonly used security means on websites today. It motivated us to analyze these schemes. The following section describes the collection of Devanagari script-based CAPTCHA that we have analyzed from a security and usability point of view.

#### 4. Data Collection

We have collected 28 different types of Devanagari CAPTCHAs. These designs are dynamic in nature. The length of these CAPTCHA schemes varies from 4 to 8 characters. The developer of these schemes has also created CAPTCHA schemes with numerals. We have not considered the numerical symbol-based schemes. There are two reasons for this. First, the numerals are not Devanagari script-based. Second, the Roman numeric characters are recognized with very high success rates in the OCR field. That makes the scheme less secure from attack. We have collected approximately 1000 samples under each CAPTCHA scheme that make the total corpus of 28000 CAPTCHAs. These samples include 39 Devanagari characters. The length of these schemes varies from 4-6 characters. Only alphabets are used in all these schemes. The size of the CAPTCHA image is 250×40. All these collected samples are highlighted in Table 1. The Sample field shows the actual image of CAPTCHA, the Scheme field represents the name of the scheme given in this paper, and the Type field contains the color depth and aspect ratio of the sample image.

**Table 1:** Collected samples of Devanagari CAPTCHA

S.N.	Sample	Scheme	Type
1		Scheme 1	8 bits Gray Scale
2		Scheme 2	8 bits Gray Scale
3		Scheme 3	8 bits Gray Scale
4		Scheme 4	8 bits Gray Scale
5		Scheme 5	8 bits Gray Scale
6		Scheme 6	8 bits Gray Scale
7		Scheme 7	8 bits Gray Scale
8		Scheme 8	8 bits Gray Scale
9		Scheme 9	8 bits Gray Scale
10		Scheme 10	8 bits Gray Scale
11		Scheme 11	24 bits RGB
12		Scheme 12	24 bits RGB
13		Scheme 13	24 bits RGB
14		Scheme 14	24 bits RGB
15		Scheme 15	24 bits RGB
16		Scheme 16	24 bits RGB
17		Scheme 17	24 bits RGB
18		Scheme 18	24 bits RGB
19		Scheme 19	24 bits RGB
20		Scheme 20	24 bits RGB

21		Scheme 21	24 bits RGB
22		Scheme 22	24 bits RGB
23		Scheme 23	24 bits RGB
24		Scheme 24	24 bits RGB
25		Scheme 25	24 bits RGB
26		Scheme 26	24 bits RGB
27		Scheme 27	24 bits RGB
28		Scheme 28	24 bits RGB

In our collected designs, the developer has rejected many characters to make the humans very comfortable with the characters. Some of the similar characters are removed to avoid the risk of decreasing usability. Table 2 shows the various characters that have been selected or rejected by the developers while creating the schemes. The table shows used, dropped, and similar characters present in the 28 CAPTCHAs. Similar characters are used for enhancing security and while some characters are dropped to improve usability.

**Table 2:** Selected/Rejected Devanagari characters in collected schemes

Character set	Characters
<b>Included characters</b>	अ आ इ उ ऋ ल ए ओ क ख ग घ ङ च छ ज झ ञ ट ठ ड ढ ण त थ द ध न न प फ ब भ म य र र ल ळ
<b>Similar characters</b>	[ अ, आ, ओ ], [ क, फ ], [ ल, ळ, ल ], [ घ, ध ], [ ज, झ, ज ], [ ट, ढ ], [ ड, ड ], [ न, न ], [ प, थ, य ], [ भ, म ], [ र, र ]
<b>Dropped characters</b>	ई, ऊ, ऐ, औ, व, श, ष, स, ह

It is clear from Table 2 that a total of 39 characters are used in each scheme. The following Table 3 displays the number of characters under each class for each CAPTCHA. We decided to have a collection of at least 50 characters under each character class under each CAPTCHA scheme. That makes a total of 2,000 (approximately) characters under each CAPTCHA scheme. It makes a total of 56,000 character images in all 20 schemes.

**Table 3:** Total characters in collected schemes

Letters	Scheme																											
	SCHEME_1	SCHEME_2	SCHEME_3	SCHEME_4	SCHEME_5	SCHEME_6	SCHEME_7	SCHEME_8	SCHEME_9	SCHEME_10	SCHEME_11	SCHEME_12	SCHEME_13	SCHEME_14	SCHEME_15	SCHEME_16	SCHEME_17	SCHEME_18	SCHEME_19	SCHEME_20	SCHEME_21	SCHEME_22	SCHEME_23	SCHEME_24	SCHEME_25	SCHEME_26	SCHEME_27	SCHEME_28
अ	52	54	50	52	50	53	50	50	51	50	51	52	50	55	50	50	50	51	50	50	50	53	51	50	52	50	50	53
Total	1430																											

त	ग	ढ	ड	ठ	ट	ज	झ	ञ	छ	च	ङ	घ	ग	ख	क	ओ	ए	ल	ऋ	उ	इ	आ
53	54	50	50	51	54	55	53	55	55	54	53	51	50	50	53	51	53	55	53	52	54	51
51	52	50	52	51	50	52	50	52	52	50	53	50	52	53	53	55	52	54	52	53	52	53
52	52	52	54	53	51	50	50	50	50	50	53	50	56	54	52	57	52	55	54	50	52	50
50	52	50	50	52	53	52	50	52	50	53	54	53	53	52	52	53	52	51	52	53	52	53
52	53	50	54	53	55	50	52	50	54	53	53	50	52	50	50	52	52	51	53	53	52	53
56	50	52	54	57	52	55	54	53	55	57	54	55	55	50	55	52	53	56	55	54	53	52
50	50	50	50	50	55	50	53	52	50	52	55	50	52	50	53	50	50	54	54	50	50	51
50	50	50	50	50	50	50	50	50	50	50	51	50	50	50	50	50	50	50	50	50	51	53
52	53	50	51	51	56	50	50	51	51	51	50	52	54	50	55	52	50	51	50	51	50	51
50	50	50	50	50	55	54	55	50	53	50	50	51	51	50	52	50	50	50	51	50	50	50
55	50	50	51	50	51	50	50	50	50	50	51	50	50	50	51	51	51	53	50	50	50	52
50	52	53	50	54	50	50	50	51	55	53	54	50	53	55	54	53	54	53	55	54	50	50
53	56	56	54	52	50	53	50	50	50	55	50	50	51	50	50	51	50	50	55	50	50	51
52	57	55	50	54	55	53	53	53	50	51	53	52	50	51	52	54	53	54	54	55	55	54
50	50	50	50	50	50	52	53	52	50	50	54	56	55	50	53	52	51	52	53	54	51	51
55	50	53	50	51	51	55	50	53	52	52	53	54	54	54	54	54	55	55	54	53	52	51
54	50	53	56	56	53	55	55	54	53	52	51	52	50	50	50	51	51	50	50	50	50	50
56	56	55	52	53	52	51	54	51	50	50	54	50	50	52	50	56	53	50	51	51	50	52
50	55	54	52	54	52	50	50	55	50	50	50	52	51	50	53	50	54	50	53	56	56	50
50	51	54	54	53	55	50	50	50	50	51	52	51	55	50	56	52	55	50	51	50	51	56
53	54	53	50	54	53	53	55	53	52	55	55	53	54	50	50	55	54	55	54	53	54	56
56	55	50	54	55	53	50	54	55	50	56	50	55	56	54	54	55	53	56	57	50	55	52
53	54	51	52	53	53	53	53	50	54	54	53	52	53	52	51	51	50	50	50	52	52	50
52	53	51	51	51	52	51	51	52	51	53	54	50	54	53	50	54	54	52	54	54	53	54
52	54	53	54	52	53	50	50	54	50	54	52	54	54	53	54	53	50	54	55	50	53	52
54	50	53	52	51	50	54	50	54	50	53	56	56	55	52	53	55	50	50	53	53	52	55
50	50	50	50	51	51	51	50	51	52	52	55	50	52	52	50	55	54	52	52	51	52	56
51	50	51	50	51	50	53	54	53	50	51	53	53	53	50	53	53	54	51	53	53	54	54
1462	1463	1449	1447	1463	1465	1445	1449	1465	1439	1462	1476	1452	1475	1437	1463	1477	1460	1464	1478	1455	1456	1463




Total	ढ	ल	३	र	य	म	भ	ब	फ	प	ज	न	ध	द	थ
2054	53	54	56	55	50	53	52	51	52	53	54	50	55	54	51
2021	50	50	53	51	50	51	53	52	50	51	52	54	53	53	52
2039	50	53	54	52	51	51	51	55	52	53	56	54	50	53	53
2020	51	52	53	52	50	53	50	50	53	53	53	50	50	52	52
2021	52	50	51	50	53	50	52	50	50	52	53	55	50	52	54
2082	54	50	53	54	54	54	53	54	52	50	50	50	53	54	55
2014	54	50	53	52	52	54	50	50	54	53	50	52	53	54	52
1974	51	50	56	51	52	50	52	51	52	50	52	51	50	50	51
2016	51	54	53	54	50	54	53	51	50	53	53	50	53	53	51
2014	55	54	53	52	52	53	53	51	52	52	54	53	52	51	50
1976	50	50	52	50	50	50	50	50	51	50	53	50	52	50	51
2042	51	53	56	50	50	50	51	54	56	50	55	50	55	56	50
2031	54	53	55	56	54	54	55	54	52	50	50	50	53	54	50
2091	54	52	57	55	52	53	54	57	53	55	54	55	56	55	54
1990	51	51	50	50	50	50	50	50	50	50	50	50	50	50	51
2043	54	52	52	51	55	52	53	50	51	51	52	50	51	52	52
2007	50	50	50	50	50	51	50	50	52	52	52	51	50	53	50
2038	52	50	50	53	53	54	53	51	54	53	52	56	54	50	53
2032	52	53	51	50	52	50	54	52	50	54	53	52	53	54	55
2036	52	51	54	55	55	50	52	53	50	55	54	51	53	54	50
2070	53	55	50	54	50	54	53	53	53	52	54	53	53	52	53
2076	50	50	52	51	50	51	51	53	56	56	55	54	50	53	56
2034	51	50	52	51	53	51	51	50	55	52	54	53	54	56	54
2034	53	50	51	54	54	53	50	53	50	51	50	54	51	54	52
2049	51	53	54	50	50	51	54	54	53	55	50	53	52	53	54
2049	50	55	54	51	53	54	50	50	51	54	54	53	55	52	53
2016	51	52	52	56	52	52	50	50	51	50	55	52	51	51	51
2048	55	52	55	55	53	51	52	53	53	55	53	54	50	53	53
56917	1455	1449	1482	1465	1450	1454	1452	1452	1458	1465	1477	1460	1462	1478	1463

#### 4.1 Techniques used for making secure and usable Devanagari CAPTCHA





Noise and distortion are two major components of CAPTCHA that decided the security and usability of the CAPTCHA. Noisy backgrounds, noisy patches, and different kinds of local and global distortions are applied to the portions of the CAPTCHA to make it secure. The collected samples are unique, and we have highlighted the global and character level noise added to designs these schemes. We also have highlighted the usability criteria that are followed in all these schemes in Table 4. The Global noise field highlights the noisy pattern applied in the background of the image; the Character level noise field represents the distortion and noise applied on the individual characters, and the Usability field represents the usability criteria followed in the scheme.

Table 4: Security and usability criteria used in designing Devanagari CAPTCHA

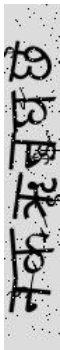



S c h e m e	Global noise	Charterer level noise	Usability
-------------	--------------	-----------------------	-----------





	Horizontal and vertical broken lines Salt noise Irregular pattern of lines Touching of lines with characters Thick and thin lines Background and foreground color are same Random gap between characters	Broken characters Random rotation of characters Multi scale of characters Overlapping of characters	Less overlapping in characters Boundary of characters makes characters visible clearly
Scheme 2 	Irregular chess pattern Irregular width of the chess boxes Color of the pattern matches with characters	Multi scaled characters Irregular characters color pattern	No overlapping and rotation in characters Noisy pattern makes the character very confusing
Scheme 3 	A shadow under each character Irregular displacement of shadow Random rotation of shadow Random gap between characters	Random rotation of characters Characters are filled with gray dots Random presence of character contour Irregular width of characters	Shadow makes the character very hard to recognize Solid white background makes the design clear
Scheme 4 	Random sized solid black circles Random location of black circles Touching of noisy patch with characters Random gap between characters	Random width of the characters Random rotation of characters Random thickness of characters	No overlapping Dots falsify the characters



<p>Scheme 5</p> 	<p>Two noisy pattern arcs and salt &amp; pepper Random position of arcs Touching of arcs with characters Random thickness of arcs Random horizontal and vertical gap among characters</p>	<p>Random aspect ratio of characters Random rotation of characters Characters are filled with pepper noisy patterns</p>	<p>Arcs are think Characters are very bright and clear</p>
<p>Scheme 6</p> 	<p>Four noisy patches Noisy patches has foreground and background color Touching of noisy patch with characters Random pattern of noisy patches</p>	<p>Broken characters Random rotation Random scaling</p>	<p>Clear background with less noise Hollow characters are easy to recognize</p>
<p>Scheme 7</p> 	<p>Three types of arc irregular pattern Three types of irregular slanting lines Background and foreground colors in noisy pattern</p>	<p>Character have same noise color Irregular horizontal and vertical placement Multi scale Random rotation</p>	<p>Color of the background is similar to characters Thickness of arcs is same as of characters</p>
<p>Scheme 8</p> 	<p>Irregular noisy pattern of sun rays Overlapping of noisy pattern with characters Thick and thin sun rays</p>	<p>Multiple shades of characters Random width of characters Random rotation of characters</p>	<p>Double color makes some characters not easy to recognize Solid character are very clear</p>







 <p>Scheme 9</p>	<p>Three noisy irregular patterns Color of the noisy patches is same as characters Touching of noisy patches with characters</p>	<p>Irregular gap among characters Random rotation</p>	<p>Touching noise is very annoying Character are misinterpreted</p>
 <p>Scheme 10</p>	<p>Two color in the background Irregular pattern of chess board Overlapping of background and foreground</p>	<p>Multi shades in single characters Irregular color pattern in characters</p>	<p>Wavy chess pattern color is overlapped on characters Characters are divided into multiple segments</p>
 <p>Scheme 11</p>	<p>Three noisy patches Touching and overlapping of noisy patches with characters Color of background is present in characters</p>	<p>Broken characters Random thickness of characters Random gap among characters</p>	<p>Characters are clear Clear difference between foreground and background color Solid background also makes the characters very distinct</p>
 <p>Scheme 12</p>	<p>Three noisy patches ( thick colored line, thin gray line and a diamond patch) Thin lines are overlapped with characters Thick line cut across the characters Touching diamond patches</p>	<p>Character are misinterpreted with diamond patches Thin lines break the characters</p>	<p>Solid background makes the appearance of characters clear No overlapping Very less dot noise Only one line as noise</p>

 <p>Scheme 13</p>	<p>Irregular displacement of shadow Different color in each time A shadow under each character Random gap between characters Random rotation of shadow</p>	<p>Random rotation of characters Characters are filled with color dots Random presence of character contour Irregular width of characters</p>	<p>Characters are very disturbing due to random size and placement of shadow White background is helpful</p>
 <p>Scheme 14</p>	<p>Two noisy pattern Irregular pattern of arcs Random colored background</p>	<p>Character are broken due to noisy thin lines Random rotation Irregular gap among characters</p>	<p>Bright solid colored background Darker characters Wide gap between arcs</p>
 <p>Scheme 15</p>	<p>Thin spiral of background color overlap with characters Random sized and shaped dots of foreground color Touching of dots with characters</p>	<p>Irregular gap among characters Random rotation and scaling Broken characters</p>	<p>Solid white background Only one colored characters Widely disbursement of dots</p>
 <p>Scheme 16</p>	<p>Thick multi-colored lines Random angles of lines Irregular pattern of lines Touching thick lines with characters</p>	<p>Multi scaled characters Multi colored characters Random rotation and gap among characters</p>	<p>To many thick lines makes design very hard Lines color is same as of characters Intersecting lines also decrease the usability</p>

<p>Scheme 17</p> 	<p>Background with same color of characters Random sized circular patches with outer glow effect Touching of circular patches</p>	<p>Character are presented with outer glow effect Random scaling and rotation Overlapping Irregular horizontal gap among characters</p>	<p>Solid background Outer glow of characters is very helpful Very less number of noisy patch</p>
<p>Scheme 18</p> 	<p>Color of background and foreground is same Randomly placed lite gray circular patches</p>	<p>Characters have rotated shadow Irregular placement of shadow Random rotation of characters</p>	<p>Only one colored background Random sized outer glow makes the characters hard to recognize dots shaped noisy patches also obfuscate the users</p>
<p>Scheme 19</p> 	<p>Multi-colored background Random colored characters are used as noisy pattern Noisy characters are touching with main characters</p>	<p>Random colored characters Overlapping of characters Random rotation Irregular gap among characters</p>	<p>Small touching characters makes the design less usable Solid bright back color is helpful</p>
<p>Scheme 20</p> 	<p>Foreground and background color is similar Thin random strokes Strokes are touching with foreground information</p>	<p>Neon colored outer glow is present Random strength of the outer glow Distorted characters with wavy patterns</p>	<p>Dark background and neon outer glow makes the characters very distinct Think noisy lines do not create much usability issue</p>

<p>Scheme 21</p> 	<p>Same foreground and background color Randomly rotated thin lines Lines are crossing over characters Color of lines is same as of characters</p>	<p>Random rotation Random gap among characters Random vertical placement Random thickness of characters</p>	<p>Clean solid background Hollow characters looks very clear Noisy lines are brighter than the characters contour color</p>
<p>Scheme 22</p> 	<p>Same background and foreground color Lines are crossing over characters Foreground colored irregular rotated thin lines</p>	<p>Distorted characters Irregular gap among characters Heavy outer glow with same color as of noise</p>	<p>Very few noisy lines High contrasted characters are easy to recognize Character distortions are easy to absorb by humans</p>
<p>Scheme 23</p> 	<p>Single colored background Two noisy patterns (a curve and polygon shaped patch) Color of noise is similar to foreground Touching of patches and curve with foreground information</p>	<p>Random aspect ratio of characters Irregular horizontal and vertical placement of characters Random scaling of characters</p>	<p>Combinations of background and foreground color is not very usable Noisy patches makes the characters very confusing Curvy patch also hides the character information</p>
<p>Scheme 24</p> 	<p>Similar foreground and background color Random sized circular colored patch Irregular pattern of thick and thin lines</p>	<p>Two different colors in characters Single character contains multiple color at irregular places Random rotation of characters Random width of characters</p>	<p>Noisy patch creates no usability issue Solid character are very clear Noisy lines are not very thick</p>

	<p>Multiple irregular number of single colored lines Irregular orientation of lines Noisy lines are cutting across the foreground</p>	<p>Random placement of characters Random rotation of characters Random sized characters</p>	<p>Too many lines decrease the usability as most of the character are hidden While solid back color is helpful but not enough</p>
	<p>Irregular lines Random orientation of lines Random colored lines Noisy lines are crossing over the characters</p>	<p>Multi-colored characters Irregular width of characters Random vertical placement Random rotation</p>	<p>Color of noisy lines is same of characters that decrease the usability Number of lines are more than needed</p>
	<p>Random color of background Random number of noisy lines Irregular placement of lines Random rotation of lines Lines are randomly colored similar to the characters Spiral patterned overlapping noise of background colors</p>	<p>Overlapping in characters Randomly colored characters Broken characters Random rotation of characters</p>	<p>Background color is not attractive Characters are widely placed Noisy lines are not decreasing usability</p>
	<p>Multi colored background Irregular patterns of colored waves with black contour Random thickness of wave contour Irregular thick lines on the inner edges Waves are cutting across the characters</p>	<p>Random thickness of characters Random rotation among characters Random gap among characters</p>	<p>Dark multi colored background character information is not clear Character are widely placed that increases the usability a little bit</p>

## 5. Points of discussion

Devanagari script-based CAPTCHAs are very useful for Indian websites. It seems good to implement these collected CAPTCHAs, yet these must be tested before deployment. So it is a dire need to test these CAPTCHAs from a usability and security point of view. We have highlighted some key points of the collected samples that generate some questions on these available designs.

## 5.1 Points related to security

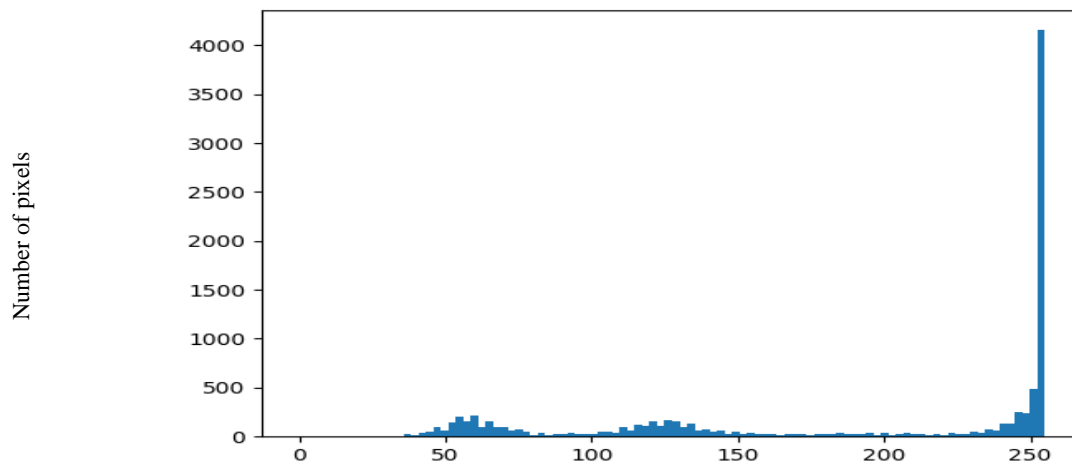
### 5.1.1 Weak design

Some of the designs are very easy to break. CAPTCHA designs with simple backgrounds (Scheme 6, Scheme 13) and a specific noisy pattern (Scheme 4, Scheme 15) are easy to break. Such CAPTCHA designs are easy to be de-noised using image processing filters. Fig. 1 shows the idea of how such schemes are vulnerable to attack. Fig. 1(a) contains a noisy shadow that is designed to obfuscate the computer, but Fig. 1 (b) has removed this noisy shadow very effectively by using histogram-based segmentation.

**Fig. 1.**  
Removing  
shadow effect



To apply histogram-based segmentation on Scheme 13, we have generated a histogram of the grey values, as shown in Fig. 2. Most of the darker values reside somewhere in the range of 100 to 150. We have decided to pick a range from 110 to 160 for de-noising this scheme. The complete algorithm for de-noising is shown in Fig. 3.



**Fig. 2.** Histogram for Scheme 13

#### **Algorithm: DE\_NOISE\_DEVANAGARI\_CAPTCHA\_13**

```

Step 1: Read the colored Devanagari CAPTCHA image
Step 2: Convert the colored image into grey scale mode
Step 3: Convert the image into a binary image using the histogram based method. (threshold value is 128)
3.1 IF pixel value is < 110 AND pixel value is < 160 THEN
    Set the pixel value = BLACK
ELSE
    Set the pixel value = WHITE
END IF
Step 4: Create a rectangular kernel matrix  $K$  of size  $3 \times 3$ 
Step 5: Perform the MORPHOLOGICAL EROSION operation twice by using kernel  $K$ 
Step 6: Save the cleaned image for character segmentation
    
```

**Fig. 3.** Algorithm for de-noising Scheme 13

### 5.1.2 Use of colors

Designs have used the concept of color without any intelligence. Designs with solid background color (Scheme 11, Scheme 12, Scheme 14, and Scheme 24) have no benefit from the security point of view. Colors can be easily removed using several image processing tools, as shown in Fig. 4. Fig. 4 (a) is a colored scheme that is nicely removed in Fig. 4 (b). The remaining noise is also removed using morphological operators in Fig. 4 (c). The algorithm for de-noising Scheme 11 is shown in Fig. 5.

Fig. 4. Removing color effect



**Algorithm: DE\_NOISE\_DEVANAGARI\_CAPTCHA\_11**

```

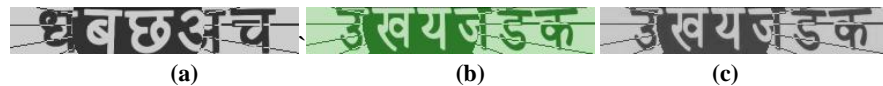
Step 1: Read the colored Devanagari CAPTCHA image
Step 2: Convert the colored image into grey scale mode
Step 3: Convert the image into a binary image using the single thresholding method. (threshold value is 128)
3.1 IF pixel value is < 128 THEN
    Set the pixel value=WHITE
ELSE
    Set the pixel value = BLACK
END IF
Step 4: Create a rectangular kernel matrix  $K$  of size  $2 \times 2$ 
Step 5: Perform the MORPHOLOGICAL CLOSING operation once by using kernel  $K$ 
Step 6: Perform the MORPHOLOGICAL OPENING operation once by using kernel  $K$ 
Step 7: Crop the image from its boundary by 5 pixels to remove any noise on the border due to the using of morphological operations.
Step 8: Save the cleaned image for character segmentation
    
```

Fig. 5. Algorithm for de-noising Scheme 11

### 5.1.3 Similar designs

Scheme 3 and Scheme 13 are similar CAPTCHA schemes. In Scheme 3, the gray scale image is used, and in Scheme 13, it is a colorful image. Scheme 8 and Scheme 24 are also very similar in nature. Other similar pairs of schemes are Scheme 21, Scheme 22, Scheme 25, and Scheme 26. After removing the colors of these schemes, they look just alike. In this case, if one scheme is compromised, then the second one is also assumed as insecure. Designs, shown in Fig. 6 (a) and Fig. 6 (b), are different in just one aspect, i.e., color. After removing the color information in Fig. 6 (b), it looks similar to Fig. 6 (a).

Fig. 6. Similar design



### 5.1.4 Noisy patch

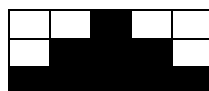
The noisy patch must not be a specific kind. In Scheme 4, a noisy patch of the solid circle is used. In Scheme 12, a diamond-shaped noisy patch is used. Scheme 15 is also following this pattern. Scheme 25 uses only lines of the same color and thickness. These kinds of noisy patterns can be removed with segmentation or coding methods, as shown in Fig. 7 (a) and Fig. 7 (b). The diamond-shaped patch in Fig. 7 (a) is completely removed in Fig. 7 (b) without disturbing the image quality.

Fig. 7. Removing Noisy patch



Fig.8 (a) shows the kernel that can be used to remove this patch using the convolution method. Fig. 8 (b) represents the algorithm to remove such a patch using the kernel shown in Fig. 8 (a). The threshold value is equal to the numeric value of the patch color.

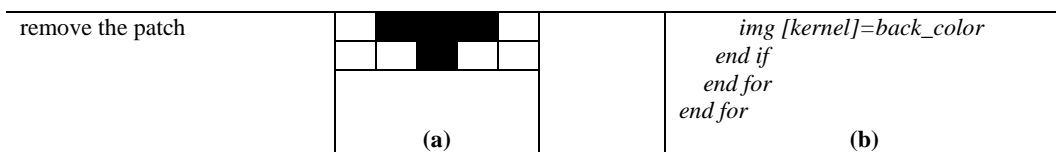
Fig.8. (a) A  $5 \times 5$  kernel for removing the diamond shaped patch, (b) Algorithm to



```

for i = 0 to row
  for j = 0 to col
    if kernel > threshold then
    
```





## 5.2 Points related to usability

### 5.2.1 Less usable

In our collected samples, some designs are less useful for all internet users. Too much noise or touching the noisy patterns makes it hard to identify the CAPTCHA. Increasing security is also the cause of it. It makes the scheme so irritating that users can leave the websites after not completing the challenge. Scheme 2, Scheme 7, Scheme 10, Scheme 13, Scheme 16, scheme 18, Scheme 20 Scheme 25, Scheme 26, and Scheme 28 are such designs that are not very easy for human users also as shown in Fig. 9 (a), Fig. 9 (b) and Fig. 9 (c).

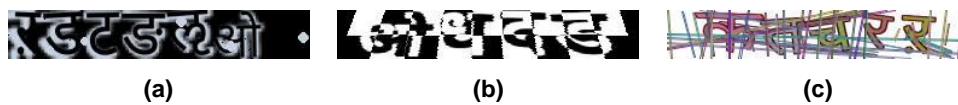


Fig.9. (a) , (b) and (c) Less usable schemes

### 5.2.2 Use of colors

CAPTCHA is meant for everyone, not for the person who has healthy eyes. Many peoples have problems like color blindness. It is also difficult to extract characters from a heavy colorful image. Scheme 16, Scheme 19, Scheme 26, and Scheme 28 are using colors heavily. It makes the design so difficult to identify that many users do not feel comfortable. Fig. 10 (a) and Fig. 10 (b) are designed with heavy use of colors in which the character color is used both in noisy patches and background. Fig. 10 (a) alphabet bet ‘अ’ looks like ‘आ’, and in Fig. 10 (b) alphabet ‘ओ’ also look like ‘आ’.

Fig.10.  
Removing  
Noisy patch



### 5.2.3 Visibility issues

Even binary color images are hard to solve due to the overlapping of background in the foreground. Scheme 2, Scheme 7, and Scheme 10 are perfect examples of such cases. Scheme 26 and Scheme 28 also have this property of overlapping colors with background and foreground. That makes it very annoying and hard to recognize for all users. CAPTCHAs, shown in Fig.11 (a) and Fig. 11 (b), are also very uncomfortable for the eyes. It is due to the same foreground and background colors and too many complex backgrounds.

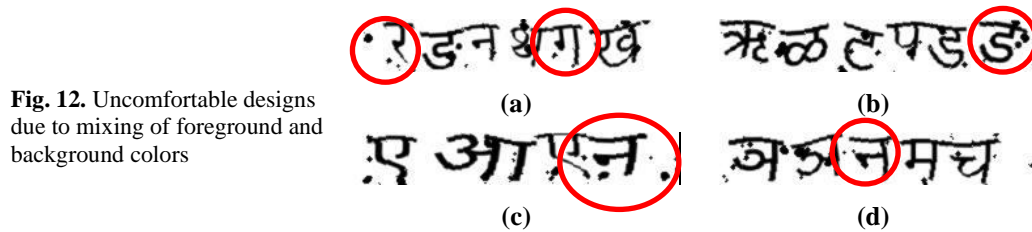
Fig. 11. Uncomfortable  
designs  
due to mixing of  
foreground and  
background colors



### 5.2.4 Noisy patch

Several issues need enough consideration while selecting appropriate noisy patch shape and color. In Scheme 4, the shape of a noisy circle is similar to the actual character. It makes the wrong interpretation of the character. The color of the noisy patch, if the patch is used as a character, must not be the same as of

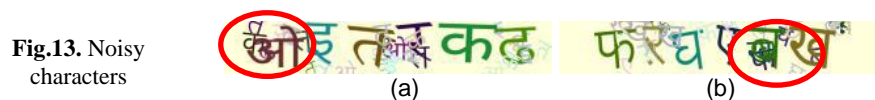
foreground. The noisy patch can be of character color if it does not play any role in the character itself. Scheme 6, Scheme 9, Scheme 12, and Schmee15 also use such noisy patches that sometimes become part of the character as shown in Fig. 12.



**Fig. 12.** Uncomfortable designs due to mixing of foreground and background colors

#### 5.2.5 Character as noise

In Scheme 19, the characters are also used as noise which makes the scheme harder to pass for many internet users. This idea fails OCR attacks, but at the same time, it decreases the usability of the CAPTCHA scheme. The character of different languages can be used as a noise. In Fig. 13 (a), it is hard to classify the highlighted alphabet. It creates ambiguity between 'र' and 'र'. Similarly, in Fig. 13 (b), the character is not identifiable as 'ब' or 'व'.



**Fig.13.** Noisy characters




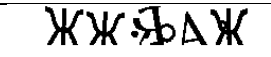







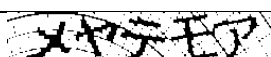


So, the above points make it very clear that design a CAPTCHA that possesses a sweet sport between security and usability is a hard problem. It is important to make a balance between the understanding of humans and computers. That is why several CAPTCHAs are available, but no one is stable for implementation. We hope that in the future, the researchers will find a better design or another alternative to CAPTCHA.

## 6. Future scope

The proposed algorithms are also able to de-noise the schemes based on other scripts. We have also tested our proposed algorithms for CAPTCHA based on other scripts and achieved a high success rate in de-noising these schemes. The results of de-nosing of these other language-based CAPTCHAs are shown in Table 5. The results look satisfactory on these scripts as well. It can be observed that the algorithms that we have developed to clean Devanagari CAPTCHAs can also be used to test the security of other language-based CAPTCHA schemes.

**Table 5.** De-noising other language CAPTCHAs

Language	Scheme	De-noised
Arabic	١٢٣٤٥٦	١٢٣٤٥٦
Arabic	٧٨٩١٢٣	٧٨٩١٢٣
Arabic	٤٥٦٧٨٩	٤٥٦٧٨٩
Arabic	١٢٣٤٥٦	١٢٣٤٥٦
Boprompfo	アキリクチ	アキリクチ
Boprompfo	LLLLLLLL	LLLLLLLL

Boprompfo		
Cyrillic		
Greek		
Greek		
Hebrew		
Hirangana		
Katakana		

## 7. Conclusions

CAPTCHA testing is not a new idea. Over the time new type of CAPTCHAs are developing and it is needed that it must be tested from security and usability point of view. Developments in the field of Computer Vision and Image Processing have made this possible to perform and deep analysis on an image. These two fields can be used to test the security of a CAPTCHA that is done in this article. We have used Image Processing to de-noise the CAPTCHA images. The results are satisfactory. Native language CAPTCHAs are developing all around the world. It is due to the increasing number of native websites that are designed for the native citizens of a country. Local government bodies also want to deliver the services to the welfare of local peoples of the country. Hackers and cyber attackers also target those websites that are used by a large population. CAPTCHA is a cost-effective solution for these types of attackers. Indian websites are now displaying the contents in local languages, and that is the reason that Devanagari script-based CAPTCHA is also available. We have analyzed 28 such CAPTCHA designs that are based on the Devanagari script. We have collected 28000 samples. We have discussed the noise and distortions used in designing these CAPTCHAs. The issues are discussed in these designs that can be benchmarks for testing the security and usability of these schemes. Any researcher who wants to do some experiments with the usability and security of these CAPTCHAs can use this article. It can be helpful as literature in this concern.

## References

- [1] Luis A, Manuel B, Nicholas J and John L.: CAPTCHA: Using Hard AI Problems for Security, EUROCRYPT 2003, LNCS 2656, 294–311 (2003).
- [2] Chew M and Baird H.: Baffle Text: a Human Interactive Proof. Proceeding of the SPIE, 5010:305-316 (2003).
- [3] Chow R, Golle P, Jakobsson M, Wang L and Wang X.: Making CAPTCHA Clickable. Proceedings of the 9th workshop on Mobile computing systems and applications, 91-94 (2008).
- [4] Chellapilla K, Larson K, Simard P and Czerwinski M.: Designing Human Friendly Human Interaction Proofs (HIPs). Proceedings of the SIGCHI conference on Human factors in computing systems, 711-720 (2005).
- [5] Ramaiah C and Govindaraju V.: A Sigma-Lognormal Model for Character Level CAPTCHA Generation. Proceedings of 13th International Conference on Document Analysis and Recognition, 966-970 (2015).
- [6] Alsuhbany S.: Optimizing CAPTCHA Generation. Proceedings of 6<sup>th</sup> International Conference on Availability, Reliability and Security (ARES), 740-745, (2011).
- [7] Imsamai M and Phimoltares S.: 3D CAPTCHA: A Next Generation of the CAPTCHA. Proceedings of International Conference on Information Science and Applications (ICISA), 1-8, (2010).
- [8] Susilo W, Chow Y and Zhou H.: STE3D-CAP: Stereoscopic 3D CAPTCHA. International Conference on Cryptology and Network Security CANS 2010: Lecture Notes in Computer Science, 6467:221-240, (2010).
- [9] Rusu A, Thomas A and Govindaraju V.: Generation and use of handwritten CAPTCHAs. International Journal of Document Analysis and Recognition, 13:49-64, (2010).
- [10] Mori G and Malik J.: Recognizing objects in adversarial clutter: Breaking a visual CAPTCHA. Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1-134, (2003).
- [11] Moy G, Jones N and Harkless C.: Distortion Estimation Techniques in Solving Visual CAPTCHAs. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1-6, (2004).

- [12] Chellapilla K and Simard P.: Using Machine Learning to Break Visual Human Interaction Proofs (HIPs). Proceedings of the Advances in Neural Information Processing Systems, 265–272, (2004).
- [13] Ahmad A and Yan J.: CAPTCHA Color, Usability and Security. IEEE Internet Computing, 16(2):1089-7801, (2012).
- [14] Gao H, Wang W and Qi J.: The Robustness of Hollow CAPTCHAs. Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security, 1075-1086, (2013).
- [15] Nguyen V, Chow Y and Susilo W.: On the Security of Text-Based 3D CAPTCHAs, Computer and Security, 45:84-99, (2014)
- [16] Gao H, Yan J and Cao F.: A Simple Generic Attack on Text CAPTCHAs. Proceedings of Network and Distributed System Security Symposium (NDSS), 1-14, (2016).
- [17] Tang M, Gao H and Zhang Y.: Research on Deep Learning Techniques in Breaking Text-Based Captchas and Designing Image-Based Captcha, IEEE Transactions on Information Forensics and Security 5(10),2522 – 2537, (2019).
- [18] Parvez M & Alsuhibany S.: Segmentation-Validation based Handwritten Arabic CAPTCHA Generation, Computers & Security, 95, 1018-29, (2020).
- [19] Yu J, Ma X and Han T.: Usability Investigation on the Localization of Text CAPTCHAs: Take Chinese Characters as a Case Study. School of Media & Design, Shanghai Jiao Tong University, Shanghai, China, ResearchGate, 1-23, (2016).
- [20] Saini B and Bala A.: Bot Protection Using CAPTCHA: Gurumukhi Script. International Journal of Application on Innovation in Engineering & Management, 2(5):267-275, (2013).
- [21] Yalamanchili S and Rao K.: A FrameWorkFor Devanagari Script-Based CAPTCHA. International Journal of Advanced Information Technology, 1(4):47-57, (2011).
- [22] Kumar M, Jindal M, and Kumar M: A Novel Attack on Monochrome and Greyscale Devanagari CAPTCHAs. ACM Trans. Asian Low-Resour. Lang. Inf. Process. 20(4), (2021)