Design and Generation of Devanagari Script CAPTCHA: Imaginative Technique

Sanjay E. Pate and R. J. Ramteke(✉)

School of Computational Sciences, Kaviyatri Bahinabai Chaudhary North Maharashtra University, Jalgaon, MS, India
rakeshj.ramteke@gmail.com

Abstract. CAPTCHA (Completely Automated Public Turing Tests to Tell Computers and Humans Apart). Only humans can pass this test, but existing computer systems cannot. It's used in a variety of machine and human identification applications. The most common type of CAPTCHA used on websites is text-based. This protected CAPTCHA script is largely made up of English letters, making it difficult for rural people who only speak their local languages to pass the test. Devanagari characters feature more sophisticated characters than typical English characters and numeral-based CAPTCHAs, considerably increasing the challenge of machine recognition. In India, most government websites present information in the Devanagari language. However, Devanagari CAPTCHAs are not utilized on websites. Therefore, we have designed a new Devanagari script text-based CAPTCHA in this article. There are 33 different varieties of Devanagari CAPTCHA images, of varying lengths (5 to 7), generated using printed and handwritten Devanagari characters and numeral combinations. Using digital image processing techniques, general rules for CAPTCHA generation are utilized to introduce noise to the CAPTCHA image so that it is not recognized or broken. The generation of a single CAPTCHA image requires 1.08 ms and 8 KB of storage. A dataset of 1,10,000 (one million and ten thousand) CAPTCHA images was created, requiring storage of 964 MB.

Keywords: CAPTCHA · Devanagari · BOTs

1 Introduction

Information security is now a key issue around the world. End-users of the Internet improved gradually. General uses of the Internet include searching, e-mail, social networking, e-banking, e-governance, etc.

Internet users have gradually improved. The Internet is generally used for things like searching, email, social networking, e-banking, and e-governance.

Universally, security and authentication are key issues. We are familiar with hackers on the internet who are ready to acquire our valuable information through bots, spammers, and dictionary attacks while operating online accounts, which creates serious issues?

R. Manza et al. (Eds.): ACVAIT 2022, AISR 176, pp. 356–381, 2023.
https://doi.org/10.2991/978-94-6463-196-8_28

Bots and spammers are heavily automated programs that create online accounts without any permission, which unnecessarily increases the damaged space. Resolution is the use of CAPTCHA.

CAPTCHA is a completely automated public Turing test to tell computers and humans apart [21]. Louis Von Ahn et al. proposed CAPTCHA in 2003. CAPTCHA protects websites and online free services (banking, etc.) against bots.

A CAPTCHA is made up of a series of alphabets or numbers that are linked together in a certain order. Random lines, blocks, grids, rotations, and other sorts of noise have been used to distort this image. A human being will have minimal trouble recognizing the set of characters. Computers, on the other hand, should ideally reject this test.

CAPTCHAs are a type of Human Interaction Proof (HIP). This process involves one computer asking a user to complete a test. The CAPTCHA test normally consists of alphabetic characters, numerals, images, or audio. Any user entering a correct response is accepted as a human, and any user who fails to enter the correct response is determined as a robot [21]. If a user fails the test, they will be refused access to the website. English characters and digit-based CAPTCHAs are often used in many applications. In this paper, we have designed a Devanagari CAPTCHA. It's a new Imaginative technique.

The following is how the rest of the paper is organized: The Study of Devanagari Script, a Literature Review on the Different Types of CAPTCHA, a Review of non-English language CAPTCHA research papers, and a review of CAPTCHA generation papers are covered in Sect. 2. Section 3: Devanagari Printed and Handwritten Alphabet and Digit Character Set, Classes of Character Set, Image Processing Techniques Used for Design, CAPTCHA Generation Guidelines, Algorithm for CAPTCHA Generation, and Flow Chart for CAPTCHA Creation are all described in Sect. 3. Section 4 discusses the hardware and software requirements for implementation. Section 5 covers the results and data set, and Sect. 6 closes the paper with a conclusion and future implications.

2 Previous Research Work

2.1 Study of Devanagari Script

The Devanagari script is the foundation of various Indian languages, including Hindi, Sanskrit, Konkani, Marathi, Nepali, Sanskrit, Dogri, Mathili, and Sindhi, among others. Devanagari is a phonetic script that originated from Ancient Brahmi. It links the sounds of alphabets to specific shapes. Except for Urdu, all of these languages are written from left to right. According to data from the year 2022, the Devanagari Hindi script is spoken by over 342 million people worldwide and ranks third among the top 45 languages [20].

Characteristics of the Devanagari Script

There are approximately 11 vowels and 33 consonants in the Devanagari script. The Devanagari script has no upper-or lower-case letters and is written from left to right. A horizontal line connecting the tops of the characters in a word is frequently used to identify it as "Shiro-Rekha". However, not all of the characters are connected in some

way. The difficulty in handwritten recognition is primarily due to the wide range of individual writing styles.

Devanagari CAPTCHA's Importance

The government of India launched the "Digital India" campaign on July 1, 2015, to make government services available to residents electronically, develop digital infrastructure to empower rural areas, notably Indian farmers, and promote digital literacy. The National Agricultural Market (e-Nam) is India's online agricultural commodity trading platform. The Indian government's goal is to build high-speed internet networks in rural areas [13].

The content of most Indian government websites is available in Devanagari script languages such as Hindi, Marathi, Haryanvi, and Gujarathi [11].

However, to protect its contents from being misused by unauthorized computer bots, the user is required to complete a CAPTCHA test. This protected CAPTCHA script is largely made up of English letters, making it difficult for rural people who only speak their local languages to pass the test. As a result, to improve the website's usability and provide easy access to native users, the CAPTCHA test must be designed in their languages, which are based on the Devanagari script [1].

2.2 Types of CAPTCHA

1. Text-based 2. Image-based 3. Audio-based 4. Video-based [11].

1. Text-based CAPTCHA:
   It is the most popular type. Background color and distortion and font style are different so that is not recognized by OCR.
   Properties of text-based CAPTCHA are a. Font b. Character set c. Distortion d. Tilting e. Waving.
   Example: ReCAPTCHA, Gimpy, EZ-Gimpy, Bongo, MSN Captcha, Baffle Text, etc. [11] (Table 1).

2. Image-based CAPTCHA: Require users to identify labeled images or rotated images (Table 2).

3. Audio-based CAPTCHA (Table 3):

4. Video-based CAPTCHA (Table 4):

CAPTCHA methods are divided into two groups:

1. OCR based: The user is shown with a distorted representation of the word and asked to type it in. Words can be recognized and detected by a human user.
   Example: Hotmail, Gimpy, Paypal, Persian/Arabic, EZ-Gimpy, Pessimal Print, Scatter type, MSN, Baffle text.
2. Non-OCR based:
   i. Visual Non-OCR based: PIX, Single Click Captcha, Collage Captcha, ASIRAA

Table 1. Text-based CAPTCHA

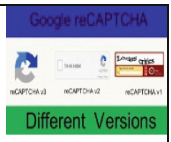
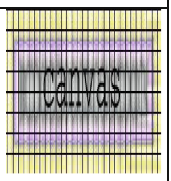
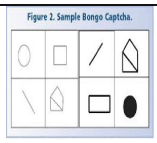
ReCAPTCHA: The reCAPTCHA service asks users to complete on-screen words shown in distorted text pictures and click on "I'm Not a Robot" using the CAPTCHA interface.[11]	
Gimpy: Instead of using automated technologies, Gimpy is founded on the idea that individuals read damaged and corrupted words. It functions by choosing words from a dictionary and showing them to the user in a distorted and corrupted visual form. The user is then prompted to type the words they saw on the screen. works in conjunction with Yahoo.[11]	
EZ-Gimpy: Instead of automated programs that operate by selecting a single word from a dictionary and then making it appear in a corrupted and distorted image format before asking the user to type the term presented in the distorted image format, EZGimpy CAPTCHA bases its operation on the idea that humans can read distorted, textured backgrounds and overwhelmed text.[11]	
Bongo: Bongo is a CAPTCHA designed to address the problem of visual pattern recognition in humans. It displays two separate block series (left and right). The	
user is tasked with identifying the feature that indicates the difference between two blocks. [11]	
MSN -CAPTCHA: Eight (upper case) characters and digits are used. Dark blue makes up the foreground, while grey makes up the background. It i s employed to produce the ripple effect and bend the characters. [11]	
Baffle -Text: At California University in Berkeley, Henry Baird creates the design. It is an altered form of Gimpy. In the Case of Baffle text, alphabets or characters are chosen at ra ndom and combined to create a pronounceable text. After that, the user is prompted to type the right term .[11]	

Table 2. Image-based CAPTCHA

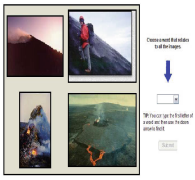
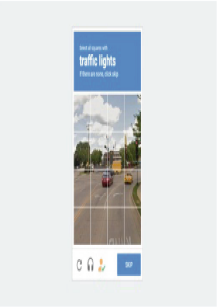
ESP-Pix: created by the reCAPTCHA team and Luis von Ahn. Four warped photos were shown to the user, and they were asked to identify them. [11]	
PIX: It is an application that has a large database of photos that have been indexed and annotated. These photos are photographs of real-life objects (a person, an animal, a flower, etc.) that have been distorted and then introduced to the viewer, followed by a question "What type of images do you have here? or identify which object is included in these images."[11]	
Asirra: It requests that the user clicks on a random image of any chosen random object from the image database.[11]	

Table 3. Audio-based CAPTCHA

Nancy Chan created the first audio-based CAPTCHA as a substitute for text-based CAPTCHAs for those with visual impairments. [11]	

ii. Non-Visual method: Audio-based Captcha.

Advantages and Disadvantages of CAPTCHA Techniques
See Table 5.

2.3 Review of Non-English Language Captcha Research Papers

1. M. Hassan Shirali-Shahreza and Mohammad Shirali-Shahreza proposed a “Multilingual CAPTCHA”. The proposed scheme is to recognizer and selects the correct image

Table 4. Video-based CAPTCHA

NuCAPTCHA is a ground-breaking CAPTCHA system that uses video to verify that users are real people and not automated programs.	

Table 5. Merits and Demerits of CAPTCHA techniques

S. No	Type of CAPTCHA	Advantages	Disadvantages
1	Text-based CAPTCHA	1. Implementation is easy. 2. Baffle Text-based CAPTCHA is used to defeat dictionary attacks. 3. Re-CAPTCHA uses new words in the dictionary that cannot be read by OCR.	1. The user has some problem identifying the correct text or characters i. Multiple fonts. ii. Font size. Iii. Blurred Letters iv. Wave Motion [11]. 2. OCR methods can Readily identify it.
2	Images-based CAPTCHA	1. Increases Security than text-based CAPTCHA. 2. An easy system based on clicks, so no need to type. 3. Image recognition pattern is a challenging AI program.	The problem of image identification for those who have low vision or are due to the blurring of images [11].
3	Audio-based CAPTCHA	1. It is used for people that have a visual impairment. 2. Friendly to people.	1. Language support 2. The character that has a similar sound.
4	Video-based CAPTCHA	1. Using OCR (Optical Character Recognition) it cannot crack. 2. Provides greater security.	Because of the big size of the file, users have trouble downloading video and finding the right CAPTCHA
5	Puzzle based CAPTCHA	1. It appears to be enjoyable. 2. It helps monitor the brain of the user.	The job is not simple for users because it takes longer to fix the puzzle-based CAPTCHA.

from among some other images displayed in a grid. Here user interfaces messages "To identify <object name>" are displayed in the user's native language so-called Multilingual. In this method, the user selected his native language than the message shown in the selected language. This strategy can effectively fend off computerized attacks since there are two steps (identification of the object and discovering the

object), neither of which can be performed correctly by a bot or spammer. On the other side, as all messages are displayed in their native language, users can operate with this technique readily. This approach was implemented using PHP language. No keyboard is needed. It is like an implicit captcha. This method can also implement on other devices such as mobile phones, PDA (Personal Digital Assistant), and the devices which have touch screens [2].

2. M.Hussn Shirali-Shahreza (2008) proposed "Advanced Nastaliq CAPTCHA". This method uses Arabian, and Persian characters of three to eight characters in length, apply Nastaliq font, and make an image file in PNG format (Portable Network Graphics). No need to add noise or distortion to the image, due to letters being connected, more than half of letters use dots, direction right to left, so OCR programs are unable to recognize the words. Shown image to the user, Compare string, test result pass or fail. Implemented in JAVA programming language [3].
3. Sushma Yalamanchili and Kameswara Rao (August 2011) proposed a "Framework for text-based CAPTCHA using Devanagari script DevaCAPTCHA for Indian languages". It is very easy to operate for Devanagari script users. OCR for Devanagari script is developed but work in progress to recognize distorted or noisy text. Key components used in the proposed framework: Database of Devanagari script text, Query generator (generate random sample), Obfuscator (distorts text and add noise), Interface, Match response. The headline "Shiro-Rekha" unites all the distinct characters of the Devanagari script.

   To obfuscate the image, the obfuscator may remove the headline and add noise using patterns like a mosaic, arcs/jaws, and vertically overlapping over the script.

   The Obfuscator further deceives the system by employing different fonts of different sizes and variable letter spacing.

   Resistant techniques of segmentation will be used in its layout.

   Future research projects will focus on DevaCAPTCHA implementation and involvement in OCR testing initiatives relating to Indian language scripts. Another future research project is to test and recognize Devanagari handwriting [4].
4. S. Ravi Kiran, and Y. Rama Krishna (2012), presented a new approach to protecting users' passwords against spyware attacks in the paper "Combining CAPTCHA and Graphical Passwords for User Authentication". In this paper, the researcher has to only recognize registered CAPTCHA images and graphical passwords to resist spyware attacks. This research is expected to advance the development of graphical passwords from a security standpoint. A Future scope is to increase improve login time and memorability [6].
5. Bilal Khan et al. (2013) presented the paper "Cyber Security Using Arabic CAPTCHA Scheme". The proposed scheme uses Arabic letters and font types to generate an image, a database of selected 2,71,0000 Arabic words, randomly select four to nine characters, create an image of letters in the rectangle with white background and blue foreground color, add light blue color dots and the line also added salt and pepper as a noises make it extremely hard for the OCR to separate noise from the image, display the image to the user for recognition. Uses preprocessing, segmentation, and character recognition techniques. The user has no trouble interacting with the system, and the algorithm is effective. According to the study, a high overall readability rate of the images was determined by a test of 150 people. So this proposed CAPTCHA

scheme can be used in Non-Arabic speaking countries where languages use Arabic scripts such as Urdu, Pashto, Persian, etc. for protecting internet resources [7].

6. Baljit Singh Saini and Anju Bala (May 2013) proposed a "Bot Protection using CAPTCHA: Gurmukhi Script". In this research paper, the proposed scheme is CAPTCHA consists of a sequence of characters in Punjabi font to generate an image. The foreground and background colors are used to distort the image and make the CAPTCHA image attractive by adding noise in the background in the shape of dots. The image is shown to the user. A User enters characters shown with each character there is an appropriate sound attached to it. Surveyed with fifty participants, the average success rate was 75%. This scheme was implemented in C# with the ASP.NET platform. The suggested approach can help secure online resources and is advantageous in countries where Punjabi is spoken. The Future scope is handwritten Punjabi CAPTCHA [8].
7. M.Tariq Bandy, Shafila Afzal Sheikh, Proposed "A model for Indian Regional Multilingual CAPTCHA challenges". In India, there are 22 official languages. Most government websites provide content in the regional language but CAPTCHA is in the English language. The researcher proposed model – i. Get the language ii. Fetch Unicode character set iii. Generate string iv. Add noise (character distortion, cluttering, deformation background, wrapping, dilation, font type, size, color) v. store string vi. Compare – test result pass or fail. Use of handwritten string/words as his future research work [9].
8. Asadullah Kehar, Rafaqat Hussain Arain et al., proposed "Design, and Development of Sindhi Text-Based CAPTCHAs for Regional Websites". In this paper Authors designed a CAPTCHA in Sindhi language, similar to the Arabic language written from right to left, containing 62 characters, Colored ellipses and clutter were inserted. Dots are used in characters so noise is not used. Implemented Sindhi Text CAPTCHA image using C# programming language, Overlapping characters were deliberately used so that CAPTCHA OCR programs could hardly segment the string. Characters string random 3 to 8 characters. Web page created using ASP, and JSP and tested by users [19].

Comparative Analysis of Research Papers on Non-English Language CAPTCHA
See Table 6.

2.4 Review of Captcha Generation Papers

S. A. Alsuhibany and M. T. Parvez, "Secure Arabic Handwritten CAPTCHA Generation Using OCR Operations," [12], In this paper Arabic Handwritten CAPTCHA generated from prewritten Arabic 123,200 part of word (PAW) images extracted from KHATT database. The proposed CAPTCHA generation method takes a PAW image *p*, a). During the segmentation process, image p is transformed to binary image c b). c is segmented. The centroid of the segmentation spots is used to estimate the baseline of the PAW body, and the segments are then slightly offset from the baseline to distort the segmentation locations. Many Arabic OCR algorithms extract features based on an assessment of the baseline since Arabic words are printed on a baseline. Therefore, if character segments in

Table 6. Comparative Analysis of research papers on non-English language CAPTCHA

SN	Name of Researcher and Year of Publication	Research paper Title	Language	Proposed Method	Distortion/Noise	Implementation/ Security	Success rate
1	M. Hassan Shirali-Shahreza and Mohammad Shirali-Shahreza (2007) [2]	Multilingual CAPTCHA	Multilingual-7 languages Dutch, French, German, Italian, Portugues and Spanish	The recognizer selects the correct image displayed in a grid. User interfaces messages are displayed in the user's native language		PHP language Resist attacks because recognition is not possible for bots	–
2	M.Hussn Shirali-Shahreza (2008) [3]	Advanced Nastaliq CAPTCHA	Arabian, Persian	String size 3 to 8 characters Uses Nastaliq font, makes an image file in PNG format, User type the word shown in the image, Test result	Colorful background with random lines, Due to letters being connected, more than half of letters use dots, direction right to left, so no need to add noise or distortion to the image.	JAVA language	Most difficult to recognize by OCR
3	Sushma Yalamanchili and Kameswara Rao (August 2011) [4]	Framework for text-based CAPTCHA using Devanagari script DevaCAPTCHA for Indian languages	Devanagari	Devanagari text Database, Random selection of string, Interface, Match response.	Font Variation, Font size, Overlapping character, Shadow Character, background, removal of shirorekha, stretching, and compression of character. Skew	Proposed as a Future work.	Not Implemented
4	S. Ravi Kiran, Y. Rama Krishna (2012) [6]	Combining CAPTCHA and Graphical Passwords for User Authentication	English	Only recognize a pre-registered combination of CAPTCHA and graphical password images	No use	Claim – Resist spyware attack	Not Mentioned

(*continued*)

Table 6. (*continued*)

SN	Name of Researcher and Year of Publication	Research paper Title	Language	Proposed Method	Distortion/Noise	Implementation/ Security	Success rate
5	Bilal Khan et al. (2013) [7]	Cyber Security Using Arabic CAPTCHA Scheme	Arabic 28 Letters	Word written right to left and number left to right, Character is of different size & shape Preprocessing, Segmentation, Character Recognition	Font type, size, vary several characters Background noise arc, lines, dots, clutter. Salt and pepper noise added	Implemented In VB.Net OCR is not developed so secure against brute force attack	Claim – readability rate best. Success rate 99.6%
6	Baljit Singh Saini, Anju Bala (May 2013) [8]	Bot Protection using CAPTCHA: Gurmukhi Script	Punjabi	a string of characters in the Punjabi alphabet that creates a picture. The user is shown the image. A user types the characters that are shown, and each character has a corresponding sound.	By adding noise in the background in the form of dots; foreground and background colors are selected	C# with ASP.NET platform	Success rate 75%.
7	M.Tariq Bandy, Shafila Afzal Sheikh (Dec. 2013) [9]	A model for Indian Regional Multi-lingual CAPTCHA challenges	Multilingual	i. Get the language ii. Fetch Unicode character set iii. Generate string iv. Add noise v. store string vi. Compare – test result pass or fail.	Character Distortion, Cluttering, Deformation background, Wrapping, Dilation, font type, size, color	Proposed Model	–
8	Asadullah Kehar, Rafaqat Hussain Arain et al. (June 2021) [19]	Design and Development of Sindhi Text-Based CAPTCHAs for Regional Websites	Sindhi	52 Sindhi characters, like Arabic, written from right to left,	Colored ellipse and clutter were inserted. Dots are used in characters so noise is not used.	Implemented interface using ASP, JSP, PHP, PYTHON	Usability tested using only Users Interface

a PAW are relocated from their baseline position, the baseline estimation approach will be more difficult. Additionally, characters in a PAW are joined at the baseline to form valleys; if characters are moved (slightly) from their intended positions, the valleys will be distorted. c) all the bounded regions are filled with random colors d) the entire image is distorted with salt and pepper noise e) random rotations are used. f) Broken characters result from some parts (going beyond the upper baseline) being drawn as dotted lines. Broken character pieces in Arabic scripts can be read correctly only in context since they can be mistaken for dots or diacritics. g) Horizontal and vertical displacements of randomly selected characters and polyline approximations.

The accuracy of CAPTCHAs is evaluated for security using a comprehensive word recognition system, with results ranging from 0.00% to 5.49%, which is very less so the researcher claims its secure design. Usability accuracy achieved more than 88%.

Mohinder Kumar1, Manish Kumar Jindal", Benchmarks for Designing a Secure Devanagari CAPTCHA", [18]. In this paper, researchers use www.captcha.com, a famous CAPTCHA designing website is used to design Devanagari CAPTCHA, where just 20 unique designs out of more than 60 available are chosen to design the CAPTCHA. A total of 39 Devanagari letters are used for the design. In each design, 2000 images were generated, and approximately 40,000 images were tested for security (breaking the CAPTCHA).

Steps for breaking the CAPTCHA are: Image converts it into a binary image, for character segmentation pre-processing is used to remove noise. When two or more characters cannot be segmented in the initial stage, post-processing is required. Technical terms included kernel sizes (morphological operations), single, double, and Otsu thresholding values, and vertical projection morphological dilation. All 20 designs have been successfully segmented and de-noised. Segmentation has a high success rate, ranging from 88.14 to 98.06 percent, although it is designed to be easily broken by bots using insecure methods.

Consequently, the researcher creates a set of recommendations for creating a secure text CAPTCHA design: Always Use Background, Similar Pattern Background must be avoided, Larger Character Set, Use of Handwritten Characters, More Use of Similar Characters, Use of Half Characters, Appropriate Use of color, Thickness of Characters, Overlapping, Scaling of Characters, Use of Font, Use of Arcs or Lines, Rotation, Length of CAPTCHA string, Vertical Position of Characters, Use of Noisy Patch [18].

V. K. Yadav, S. Agarwal, J. Uprety, and S. Batham, "SRTS: A Novel Technique to Generate Random Text," [10]. In this paper authors generate CAPTCHA text is completely based on date and time. Here system collects the current date and time, separates each digit, performs the arithmetic operation, and generates code from 62 ASCII values of English alphabets also checks the possibility of repetition of generated text and displays random text. These texts exploit the intelligence of shifting time to construct a unique sequence every time a user sends a request for a new text, making them impenetrable to unsophisticated users and standard code [10].

M. Tang, H. Gao, Y. Zhang, Y. Liu, P. Zhang and P. Wang, "Research on Deep Learning Techniques in Breaking Text-Based Captchas and Designing Image-Based Captcha," [14], In this paper authors, perform analysis of breaking text-based CAPTCHA images of different English, Chinese, Arabic, Roman language top 50 websites using deep learning technique CNN which results from 10% to 90% success rate. Here, authors employing neural style transfer techniques introduced a unique image-based Captcha called SACaptcha. While SACaptcha is based on challenges of semantic information interpretation and pixel-level segmentation, the majority of early image-based CAPTCHA is focused on the problem of image classification. This is a commendable attempt to increase the security of Captchas using deep learning methods [14].

P. Panwar, Monika, P. Kumar, and A. Sharma, "CHGR: Captcha generation using Hand Gesture Recognition," [16]. This research proposes a more efficient methodology for generating CAPTCHAs using hand gesture recognition techniques, as opposed to

typical CAPTCHAs. This solution involved displaying a form along with one CAPTCHA image of a hand gesture from the database and a message telling the user to copy the gesture. The user then performed a motion in front of the system's camera, and the model checked to see if it matched the gesture that had been displayed or not. As it is impossible for the robot to learn how to repeat the same move as seen in the image on its own, using this way will undoubtedly ensure the user's identification. These CAPTCHAs can be used in the future by websites to determine if a user is a human being or a robot, requiring less time and effort from humans. The experiments produce findings that are satisfactory for the suggested model. Three steps are involved in pre-processing:1. Image Conversion 2. Morphological Filtering -dilation and erosion to fetch smooth, closed, and complete contour of gesture. 3. Edge Detection- Canny edge detection algorithm has been used due to the above advantages and it eliminates the risk of multiple responses to one edge. The algorithm first smooth's the image obtained after conversion and morphological filtering to remove any noise if present. The gradient is then calculated by the algorithm to emphasize the areas with spatial derivatives, and by employing this, the maximum pixel values are suppressed. The resultant gradient array was then further decreased via hysteresis to improve accuracy. Hysteresis uses two thresholds to keep track of the pixels that haven't yet been suppressed. The pixel value is set to zero if the magnitude of the pixel is less than the smaller threshold, and to one of the greater thresholds is exceeded. C. Matching: To match the gesture made by the user to the shown as CAPTCHA, pixels were first counted contributing to the edges in the image obtained from the result of canny edge detection. The average number of pixels was then counted for each gesture and this value was further used for matching. Once the pixel count of the input image is obtained, an error has been then calculated for each gesture. We created 100 images using the database of hand gestures from 20 distinct people, each with five different gestures. The model is 80% accurate as a result [16].

Comparative Analysis of CAPTCHA Generation Research Papers
See Table 7.

Table 7. Comparative Analysis of CAPTCHA generation research papers

Sr. No	Researchers' name with Title of paper	Language, Data set	The technique used to survive OCR attacks	Accuracy	Susceptible to attack
1	S. A. Alsuhibany and M. T. Parvez, Secure Arabic Handwritten CAPTCHA Generation Using OCR Operations [12]	Arabic 123,200 part of word (PAW) images extracted from the KHATT database	1. Segmentation, 2. Displacement, 3. Bounded regions filled with random colors, 4. Salt and pepper noise, 5. Random rotations 6. Broken characters	0.00% to 5.49%	Accuracy is a maximum of 5.49%, Means secured from Bots.

(*continued*)

Table 7. (*continued*)

Sr. No	Researchers' name with Title of paper	Language, Data set	The technique used to survive OCR attacks	Accuracy	Susceptible to attack
2	Mohinder Kumar, Manish Kumar Jindal, Benchmarks for Designing a Secure Devanagari CAPTCHA [18]	Devanagari 39 Letters Uses www.captcha.com Website for design CAPTCHA 40,000 images were designed	1. Thresholding 2. Morphological dilation 3. Segmentation Vertical and Horizontal	88.14 to 98.06%	susceptible to break by bots means not secured
3	Virendra Kumar Yadav, Shantanu Agarwal et al., SRTS: A Novel Technique to Generate Random Text [10]	English 62 Letters and Digits Random Generation of Text using server date and time	No image processing technique used	Not Tested	The authors claim it's secure but not verified
4	Pooja Panwar, Monika, Parveen Kumar et al., CHGR: Captcha generation using Hand Gesture Recognition [16]	CAPTCHA image showing some hand gestures from the database, user copies the gesture in front of the camera of the system and then the model has verified whether the gesture made by the user is the same as that shown or not.	1. Image Conversion 2. Morphological Filtering 3. Edge Detection 4. Matching 5. 20 different persons, 6 Gestures of each individual thus making a total of 100 images.	80%	Very Small database used so susceptible for brute force attack

3 Devanagari Captcha Generation Proposed Model

3.1 Character Set

We chose printed Devanagari characters and a handwritten Devanagari character set for our Devanagari CAPTCHA design. Devanagari 53 image processed alphabets

Each symbol has a size of 65 × 65 pixels (Fig. 1).

Devanagari 53 Handwritten Alphabets
Handwritten 53 Devanagari alphabets: Each symbol has a size of 65 × 65 pixels (Fig. 2).

Rejected Characters from Character Set: 19
We have designed the CAPTCHA in such a way that it does not create any confusion for humans and retains its usability, so rejected 19 alphabets (Fig. 3).

Selected Character Set: Printed PREPROCESSED Characters [34]
See Fig. 4.

Fig. 1. Printed Devanagari Character Set [53]

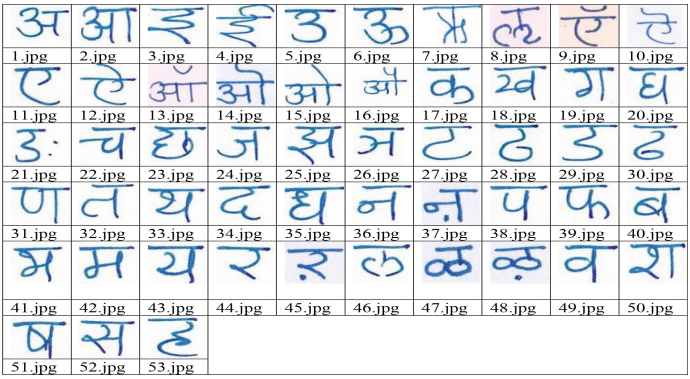


Fig. 2. Handwritten Devanagari Character Set [53]

Handwritten Characters [hc] -34
See Fig. 5.

Devanagari Handwritten 10 Numerals
See Fig. 6.

Devanagari Printed 10 Numerals: PD – 10
See Fig. 7.

Unicodes
See Fig. 8.

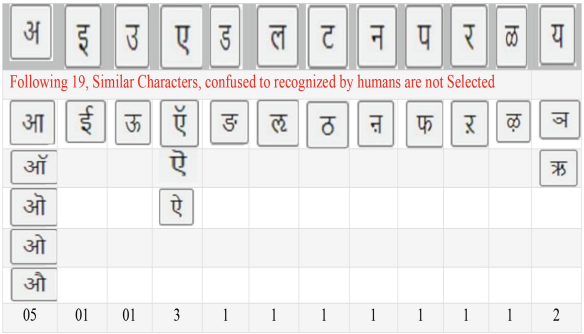


Fig. 3. Rejected Devanagari Characters [19]

Fig. 4. Selected Printed Devanagari Characters [34]

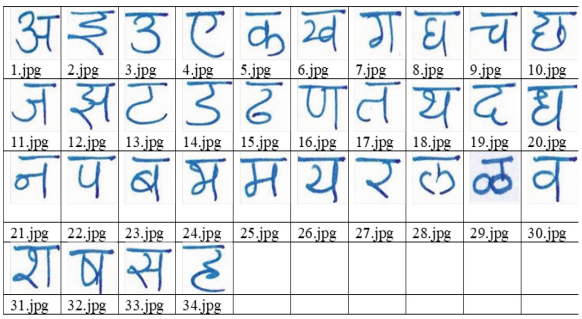


Fig. 5. Selected Handwritten Devanagari Characters [34]

	Printed	**Handwritten**
Alphabets	34	10
Digits	34	10

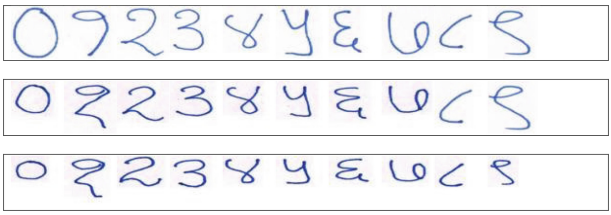


Fig. 6. Three Sets of Handwritten Numerals [10]

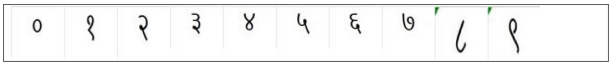


Fig. 7. Set of Printed Numerals [10]

Alphabets Unicodes
["\u0905","\u0907","\u0909","\u090F","\u0915","\u0916","\u0917","\u0918","\u091A","\u091B", "\u091C","\u091D","\u091F","\u0921","\u0922","\u0923","\u0924","\u0925","\u0926","\u0927", "\u0928","\u092A","\u092C","\u092D","\u092E","\u092F","\u0930","\u0932","\u0933","\u0935", "\u0936","\u0937","\u0938","\u0939","\u0966","\u0967","\u0968","\u0969","\u096A","\u096B", "\u096C","\u096D","\u096E","\u096F"]
Digits Unicode
["\u0966","\u0967","\u0968","\u0969","\u096A","\u096B","\u096C","\u096D","\u096E","\u096F"]

Fig. 8. Unicodes of Devanagari Characters and Numerals [34 + 10 = 44]

Table 8. Four Superclasses of charcater set

PA:Printed Alphabet-34	PD: Printed Digit-10
HA: Handwritten Alphabet-34	HD: Handwritten Digit-10

3.2 Classes of Character Set

There are **11** classes of **4** superclasses of the character set (Tables 8 and 9).

3.3 Image Processing Techniques Used

It is a technique for altering images, obtaining enhanced images, or extracting information that can be used later.

The two types of image processing techniques used are analog and digital.

Analog image processing is useful for hard copies like prints and photos.

Computer-based digital image alteration is made possible with the use of digital image processing tools.

Image Acquisition:

Table 9. 11 Classes of charcater set

PA-34	PD-10	PA-34	PD-10
HA-34	HD-10	HA-34	HD-10
PA-34	PD-10	PA-34	PD-10
HA-34	HD-10	HA-34	HD-10
PA-34	PD-10	PA-34	PD-10
HA-34	HD-10	HA-34	HD-10
PA-34	PD-10	PA-34	PD-10
HA-34	HD-10	HA-34	HD-10
PA-34	PD-10	PA-34	PD-10
HA-34	HD-10	HA-34	HD-10
	PA-34	PD-10	
	HA-34	HD-10	

Thirty-Four (34) Devanagari alphabet and Ten (10) digit samples were collected and scanned using a scanner before being converted to a picture format. (.JPG).

In addition, Thirty-Four (34) Devanagari alphabet and Ten (10) digit Devanagari handwritten samples were gathered, digitized, and transformed into image format using a scanner. (.JPG).

Image Pre-processing

I. For printed alphabets following Image preprocessing was performed:

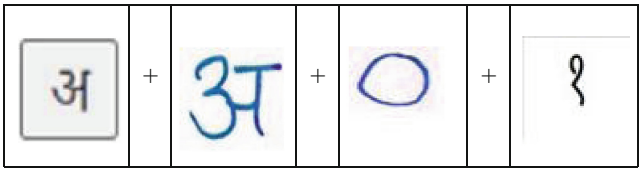
i. Each printed alphabet's background was converted using Matlab Program
ii. Each character is enclosed in a square (Image Fusion).

Resizing Image: When an image is resized or distorted from the one-pixel grid to another, it is called image interpolation. When increasing or decreasing the total number of pixels, picture scaling is required.

Using Matlab and Python programs, each printed and handwritten alphabet image is reduced to 65 × 65 pixels. (1 kb size).

Insert Noise in Image

Noise means, the pixels in the image show different intensity values instead of true pixel values that are obtained from image. []



We add noise in CAPTCHA image, because Optical Character Recognition (OCR) recognizer has difficult to break the CAPTCHA image**.**

$A(x,y) = H(x,y) + B(x,y)$

Where, A(x,y) = function of noisy image, H(x,y) = function of image noise, B(x,y) = function of original image. Following different types of noises are added to the image randomly.

Types of Noise in image:

1. ***Gaussian Noise:*** Gaussian Noise is a statistical noise having a probability density function equal to a normal distribution, also known as Gaussian Distribution. A random Gaussian function is added to the Image function to generate this noise [22].

$$p_G(z) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(z-\mu)^2}{2\sigma^2}}$$

2. ***Salt and Pepper Noise:*** Salt and Pepper noise is added to an image by addition of both random bright (with 255-pixel value) and random dark (with 0-pixel value) all over the image. Because it statistically drops the original data values, this model is also known as data drop noise [22].
3. ***Poisson Noise***: The appearance of this noise is seen due to the statistical nature of electromagnetic waves such as x-rays, visible lights, and gamma rays. The x-ray and gamma-ray sources emitted the number of photons per unit of time. In medical x-rays and gamma-ray imaging systems, these rays are injected into a patient's body from the source. The photons in these sources fluctuate at random. The resulting image is spatially and temporally unpredictable. Quantum (photon) noise or shot noise are other names for this type of noise [22].

4. **Speckle Noise:** A fundamental problem in optical and digital holography is the presence of speckle noise in the image reconstruction process. Speckle is a granular noise that occurs naturally in images and diminishes their quality. Speckle noise is produced by multiplying random pixel values with distinct picture pixels [22].

   CAPTCHA is noisy image, to break or recognise a CAPTCHA image, noise is removed using a denoising method filter to restore the original image.

3.4 CAPTCHA Generation Guidelines: [17]

1. A Larger Character Set: To construct Devanagari CAPTCHA, we used a printed and handwritten character set. There are 10,34,44,88 symbols in all.
2. Handwritten Characters are used:

3. CAPTCHA String Length: The length of the CAPTCHA string varies, ranging from 5 to 7. (A random pick is made at runtime.)
4. Characters' Dimensions (Table 10):

5. Font Selection: For the CAPTCHA design, we have only chosen ONE typeface/font. For a single printed character, we get 03 classes of CAPTCHA images.

   The number of possibilities of Devanagari CAPTCHA images produced by using more than one typeface/font, increases the complexity of the CAPTCHA image, making it more difficult to read and recognize by OCR.

Table 10. Change Symbol Image Size w.r.t. Captcha string length

CAPTCHA String Size = 5	CAPTCHA String Size = 6	CAPTCHA String Size = 7
54 X 54 Pixel size	45 X 45 Pixel size	39 X 39 Pixel size
		

6. Scaling: Resizing a digital image is known as image scaling. An image becomes smaller as it is scaled down, and larger when it is scaled up. It employs bicubic and bilinear interpolation. Scale image +5 to +10.

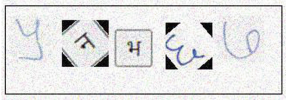
7. Rotation: Bicubic 30° and Bilinear −40° are used.

The effect of scaling and rotation on a Devanagari symbol image in a CAPTCHA drops the accuracy of CAPTCHA image recognition and breaking by OCR.
Use of lines, arcs, polygons, and rectangles:

9. Noisy Patch:

   Noise is typically defined as a random variation in brightness or color information.

   a. Increase and Decrease image contrast
   b. Added randomly any ONE of the noise to CAPTCHA Image: Gaussian, localvar, Salt and Pepper, Poisson, Speckle.



10. The symbol image has a black and white background:

3.5 Algorithm for Captcha Generation

1. PC, HC, PD, HD…..Select any one set from 11 combinations.
2. Set path of character set and CAPTCHA image directories.
3. Set the length of the character set as per the above selection [10 or 34 or 44 or 88]
4. Input required L = number of CAPTCHA images required.
5. Repeat steps 6 to 19, for CI = 1 to L
6. A: = Read square shape box image file (in which CAPTCHA text generated)
7. Randomly select length of CAPTCHA image [5 or 6 or 7] = CL
8. Set folder/directory to store CAPTCHA Image
9. Repeat step 10 to 14, for I = 1 to CL (Length of CAPTCHA is CL = 5 OR 6 OR 7)
10. B: = Read randomly anyone Devanagari character image set.
11. Resize image B
    a. if CL = 5 then 54 × 54
    b. if CL = 6 then 45 × 45
    c. if CL = 7 then 39 × 39
12. Apply any one of the five functions to the above image at random to create noise that OCR Won't recognize.
    a. Rotate image Bicubic 30 degrees.
    b. Rotate image Bilinear −40 degree
    c. Scale image + 5 points
    d. Decrease image Contrast
    e. Increase image Contrast
13. Determine the next image placement in square shape box A.
14. Insert above image B in the A (square shape box) at the calculated location.
15. Resize the image
16. Save the CAPTCHA image, just created in JPG, JPEG, or BMP format.
17. Add any noise to the Image: ["gaussian", "localvar", "poisson", "salt", "pepper", "s&p", "speckle"]
18. Insert line or circle or Polygon or ellipse at specified location in the image.
19. Save a CAPTCHA Image at the specified location.

3.6 Flow Chart

See Fig. 9.

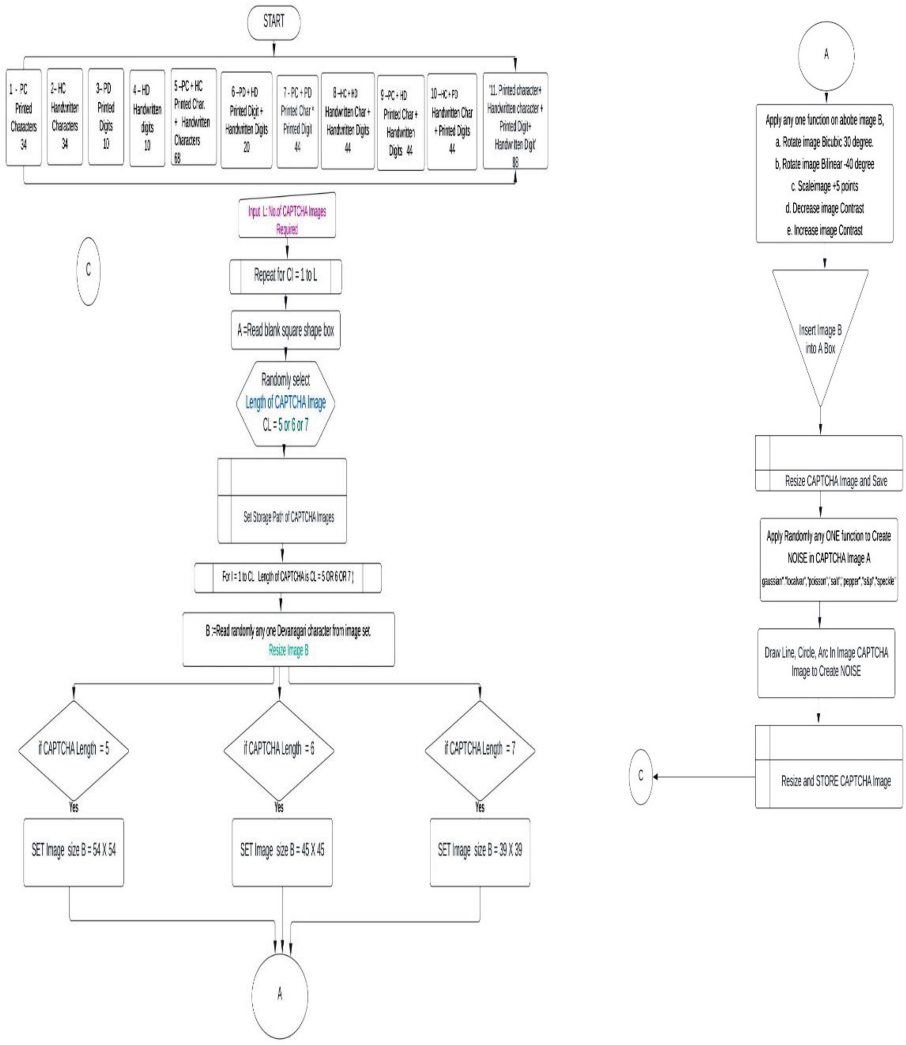


Fig. 9. Flow Chart of Design

4 Implementation

Experimental Environment

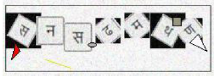
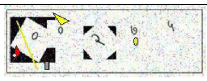
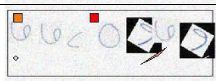
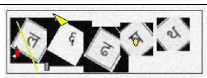
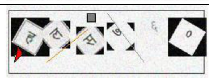
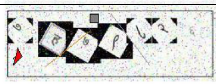
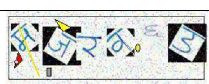
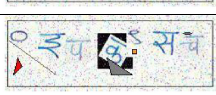
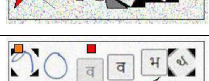
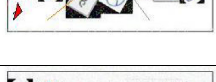
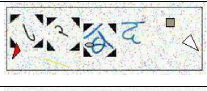
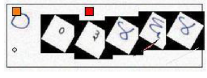
This project is implemented in the Jupyter platform in the Windows environment using Python language, with its version 3.0.0 dated 20 Feb 2020.

Computer Hardware requirement: Processor: Intel(R), Core™, i5 or new versions, CPU @ 2.20 GHz,8 GB RAM, 4 GB NVIDIA GEFORCE GTX GPU.

System type: 64-bit Windows operating system.

Software requirement: Python, Tensorflow, Keras.

Table 11. Devanagari Captcha: Generation of Printed and Handwritten Devanagari Characters with Variable Length

Sr. No	Character Set	No. of Char.	CAPTCHA Images Generated of length 5	CAPTCHA Images Generated of length 6	CAPTCHA Images Generated of length 7
1 – PC	Printed Characters	34			
2 – PD	Printed Digits	10			
3 – HC	Handwritten Characters	34			
4 – HD	Handwritten digits	10			
5 – PC + PD	Printed Characters + Printed Digits	34 + 10			
6 PC + HD	Handwritten Characters + Handwritten Digits	34 + 10			
7 – PC + HC	Printed Characters + Handwritten Characters	34 + 34			
8 – PC + HD	Printed Characters + Handwritten Digit	34 + 10			
9 – HC + PD	Handwritten Characters + Printed Digit	34 + 10 = 44			
10 PD + HD	Printed Digit + Handwritten Digits	10 + 10 = 20			
11 PC + HC + HC + HD	Printed Characters + Handwritten Characters + Printed Digit + Handwritten Digits	34 + 34 + 10+ 10 = 88			

Python requires less time to execute the code than Matlab so selected python for implementation.

For implementation used different libraries in Python.

NumPy, Pandas, Scikit-learn (Sklearn), TensorFlow, Keras, OpenCV, PyGame, PyTorch, Tesseract OCR.

OpenCV (Open Source Computer Vision library) technique used Morphing: Merging through a smooth transition different pictures to create a new one.

Used OpenCV functions read, write, display, resize, translate, scale and rotate image.

5 Result: Data Set

A total of 1,10,000 samples of CAPTCHA Images with noise were generated (Table 11).

Devanagari Captcha: Generation of Printed and Handwritten Devanagari Characters with Variable Length: Image Size 250 × 90.

6 Conclusion and Future Work

In this paper, 34 Devanagari Printed, Handwritten characters and 10 digits are used. Image acquisition, Image resize, and Image binarization these basic image processing operations are used. Digital Image processing techniques Bicubic and Bilinear interpolation are used for scaling and rotation.The effect of scaling and rotation on a Devanagari symbol image in a CAPTCHA will drops the accuracy of CAPTCHA image recognition and breaking by OCR.

Various types of noises are added to the image randomly like gaussian, localvar, Poisson, salt, pepper, salt &pepper, speckle. We add above noise in CAPTCHA image, because Optical Character Recognition (OCR) recognizer has difficulties to break the CAPTCHA image.

Standard guidelines for CAPTCHA generations are followed.04 (Four) types of Character Sets are used – Printed Alphabet, Handwritten Alphabet, Printed Digit, and Handwritten Digit. Generated 11 Classes from these 04 combinations. For each class – 03 (THREE) subclasses are created –The string length of the CAPTCHA image considered here is FIVE, SIX, and SEVEN (5, 6, 7). In total there are 11 classes × 3 subclasses = 33 subclasses. So 33 types of images were generated. For each class of selected character set, 10,000 CAPTCHA images were created.

For 11 Classes × 10,000 images = a Data set of 1,10,000 (One Million Ten Thousand) images was created using Python. For a generation of 10,000 CAPTCHA Images 180 seconds are required. So the generation of One (01) CAPTCHA image requires 1.08 milliseconds. Each image is of size 250 × 90 pixels and requires 8 KB storage. Data set of 1,10,000 CAPTCHA images created which requires 964 MB storage. Devanagari CAPTCHA is implemented using Anaconda, Jupiter platform of Python Programming language. This algorithm is helpful to design other languages CAPTCHA. Devanagari CAPTCHA is useful for all Devanagari language users and it is more secure than English CAPTCHA.

The future scope is, Design and develop a framework for recognition or breaking of Devanagari CAPTCHA image and analyze the success rate. Noise in CAPTCHA image is removed using a denoising method filter to restore the original image.

To use more than one typeface/font which increases the complexity of the CAPTCHA image, making it more difficult to read and recognize by OCR, is another future scope.

Design Audio and Video Devanagari CAPTCHA. Also, develop Devanagari CAPTCHA using Alphabets without “ShiroRekha”.

References

1. Om V. (2005). Multilingualism for Cultural Diversity and Universal Access in Cyberspace: An Asian Perspective, an Asian Perspective, UNESCO, May 2005
2. Shirali-Shahreza, M.H. &Shirali-Shahreza, M., (2007) "Multilingual CAPTCHA", ICCC 2007 IEEE International Conference on Computational Cybernetics, 19–21 October.
3. M.Hassan Shirali-Shahreza, Mohammad Shirali-Shahreza, "Advanced Nastaliq CAPTCHA", Cybernetic Intelligent Systems 2008, IEEE.
4. Sushma Yalamanchili and Kameswara Rao, "A FRAMEWORK FOR DEVANAGARI SCRIPT- BASED CAPTCHA", Cryptography and Security (cs.CR); Human-Computer Interaction, International Journal of Advanced Information Technology, Vol. 1, No. 4, August, pp. 47–57, 2011.
5. Banday M and Shah N (2011) Challenges of CAPTCHA in the Accessibility of Indian regional websites. Proceedings of the fourth annual ACM Bangalore conference, 1–4.
6. T.S.Ravi Kiran, Y. Rama Krishna, "COMBINING CAPTCHA AND GRAPHICAL PASSWORDS FOR USER AUTHENTICATION", IJRIM, Volume 2, Issue 4 (April 2012) (ISSN 2231-4334).
7. Bilal Khan, Khaled Alghathbar et al., "Cyber Security using Arabic CAPTCHA Scheme", The International Arab Journal Information Technology, Vol. No.1, January 2013.
8. Baljit Singh Saini, Anju Bala, "Bot Protection using CAPTCHA: Gurmukhi Script", International Journal of Application or Innovation in Engineering & Management (IJAIEM), May 2013, Volume 2, Issue 5. (ISSN 2319-4847)
9. M. Tariq Banday, "A Model for Indian Regional Multi-lingual CAPTCHA Challenges", 9th J&K Science Congress and Regional Science Congress, 1–3 October 2013, University of Kashmir, Srinagar.
10. V. K. Yadav, S. Agarwal, J. Uprety and S. Batham, "SRTS: A Novel Technique to Generate Random Text," 2014 International Conference on Computational Intelligence and Communication Networks, 2014, pp. 268–272, https://doi.org/10.1109/CICN.2014.68.
11. Sanjay E.Pate," A CAPTCHA: A Review ", National Conference on Recent Trends in Computer Science & Applications 2015. Date: 27 & 28 January, 2015. ISBN No. 978-93-84093-76-1(1).
12. S. A. Alsuhibany and M. T. Parvez, "Secure Arabic Handwritten CAPTCHA Generation Using OCR Operations," 2016 15th International Conference on Frontiers in Handwriting Recognition (ICFHR), 2016, pp. 126–131, https://doi.org/10.1109/ICFHR.2016.0035.
13. Seema D. (2017). Digital India Opportunities and Challenges, International Journal of Science & Technology (IJSTM) Special Issue NCIETM – 2017, 6(3): 61–671
14. M. Tang, H. Gao, Y. Zhang, Y. Liu, P. Zhang and P. Wang, "Research on Deep Learning Techniques in Breaking Text-Based Captchas and Designing Image-Based Captcha," in IEEE Transactions on Information Forensics and Security, vol. 13, no. 10, pp. 2522–2537, Oct. 2018, https://doi.org/10.1109/TIFS.2018.2821096.
15. Aldosari M (2018) Innovative multilingual CAPTCHA Based on handwritten characteristics, Intechopen, 72599.
16. P. Panwar, Monika, P. Kumar, and A. Sharma, "CHGR: Captcha generation using Hand Gesture Recognition," 2018 Conference on Information and Communication Technology (CICT), 2018, pp. 1–6, https://doi.org/10.1109/INFOCOMTECH.2018.8722409.
17. Kumar M, Jindal MK and Kumar M (2021a) A systematic survey on CAPTCHA recognition: types, creation and breaking techniques. Archives of Computational Methods in Engineering, Springer, 1–30.
18. Mohinder Kumar1, Manish Kumar Jindal," Benchmarks for Designing a Secure Devanagari CAPTCHA", Published online: 19 January 2021, part of Springer Nature 2021

19. Asadullah Kehar, Rafaqat Hussain Arain et al., Design and Development of Sindhi Text-Based CAPTCHAs for Regional Websites, SJET | P-ISSN: 2616-7069 |E-ISSN: 2617-3115|Vol. 4 No. 1 January–June 2021.
20. https://en.wikipedia.org/wiki/List_of_languages_by_total_number_of_speakers, 2022
21. L. Von Ahn, M. Blum, and J. Langford, "Telling humans and computers apart automatically," Communications of the ACM, vol. 47, no. 2, pp. 56–60, 2004
22. Charles Boncelet (2005). "Image Noise Models". in Alan C. Bovik. Handbook of Image and Video Processing.