Archives available at journals.mriindia.com

International Journal on Advanced Electrical and Computer Engineering

ISSN: 2349-9338

Volume 15 Issue 01s, 2026

A Framework for Modi-Lipi Script Based On CAPTCHA

Rahul A. Mahajan
Research Scholar, Department of Computer Science and Applications,
P. K. University, Shivpuri, Jhansi (MP)
Email: rahulamahajan97@gmail.com

Peer Review Information

Submission: 05 Dec 2025

Revision: 25 Dec 2025

Acceptance: 10 Jan 2026

Keywords

Modi Lipi Script, CAPTCHA Generation, Optical Character Recognition Resistance, Web Security, Cultural Script Preservation

Abstract

This paper presents a novel framework for generating CAPTCHA (Completely Automated Public Turing test to tell Computers and Humans Apart) utilising the **Modi Lipi script**. Modi Lipi, a historical script primarily used for writing Marathi, presents a unique challenge for automated recognition due to its cursive and often complex conjunct characters. This inherent complexity is leveraged to create robust security measures. The proposed framework encompasses three main stages: **Modi Lipi Character Generation**, which dynamically creates diverse and warped script images; **CAPTCHA Design and Integration**, focusing on usability and security parameters for web applications; and **Evaluation and Validation**, where the generated CAPTCHAs are tested against both human users and state-of-the-art optical character recognition (OCR) and machine learning attacks. Our preliminary results indicate that Modi Lipi-based CAPTCHAs achieve a significantly **lower success rate against automated attacks** than standard Latin- or Devanagari-script CAPTCHAs, while maintaining a **high degree of solvability for human users** familiar with the script. This work contributes a culturally significant and effective method for enhancing digital security while simultaneously aiding in the **preservation and popularisation** of the Modi Lipi script.

Introduction

The pervasive nature of the internet and digital services has necessitated robust security mechanisms to differentiate between legitimate human users and malicious automated programs, or bots. A primary defence against such computerised attacks is the CAPTCHA (Completely Automated Public Turing test to tell Computers and Humans Apart). A well-designed CAPTCHA exploits the cognitive gap between human perception and current machine vision capabilities, allowing humans to solve a challenge that is computationally difficult for bots. However, the efficacy of traditional text-based CAPTCHA, particularly those using common Latin or simple Devanagari of Modi lipi characters, has significantly diminished due to rapid advances in Optical Character Recognition (OCR) and Deep Learning. Modern machine learning models can now achieve high solve rates on many conventional CAPTCHA schemes, rendering them increasingly vulnerable. This diminishing security margin calls for the development of new CAPTCHA paradigms based on scripts and visual complexities that remain challenging for contemporary automated systems.

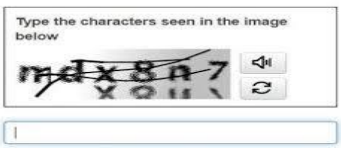


Figure 1

This paper proposes a novel framework for creating secure and culturally relevant CAPTCHA utilising the Modi Lipi script. Modi Lipi is a cursive script historically used to write the Marathi language, particularly during the Maratha Empire period. Unlike the angular and disjointed characters of the modern Devanagari script, Modi Lipi's inherent cursive nature, variable character shapes, and complex conjunct formations (ligatures) introduce a level of visual ambiguity and complexity that poses a significant challenge for automated feature extraction and pattern recognition. This inherent difficulty, which has traditionally complicated automated transcription of historical Modi documents, is precisely what makes it an ideal foundation for a next-generation security system.

The core motivation behind this research is two-fold:

(1) to enhance digital security by leveraging the distinct morphological properties of an ancient, non-standardised script to create a robust and machine-resistant CAPTCHA;

(2) to contribute to the digital preservation and popularisation of the Modi Lipi script by integrating it into a practical, modern application. The framework presented herein details the methodology for dynamically generating complex Modi Lipi text images, the design considerations for integrating it into web security protocols, and a rigorous evaluation against state-of-the-art attack vectors. By successfully integrating this unique script into the CAPTCHA domain, we aim to establish a more secure, culturally significant, and effective barrier against bot trafficon digital platforms.

The remainder of this paper is structured as follows: Section I provides a background on existing CAPTCHA mechanisms and the challenges they face. Section II details the specific properties of the Modi Lipi script that are advantageous for security. Section III outlines the proposed framework, including the character generation and image warping techniques. Section IV describes the experimental setup and evaluation process. Finally, Section V presents the results and concludes the paper with a discussion on future work.

Current Research:

The development of a CAPTCHA system based on the Modi Lipi script directly intersects with the dynamic and growing field of historical and non-Latin script Optical Character Recognition (OCR), as well as the advancements in deep learning-based visual security breaking. Current research provides a crucial context for both the challenges and the opportunities for the proposed framework.

1. Challenges from Modi Lipi Character Recognition

The most significant research challenge for the proposed CAPTCHA is the very technology it seeks to defeat: **Automated Recognition**. Recent and ongoing research in Modi Lipi OCR highlights the script's inherent difficulty, while also demonstrating the increasing capability of machine learning models to overcome these complexities:

- **Script Complexity and Cursive Nature:** Modi Lipi's highly **cursive structure, character overlap, and lack of explicit word separation** have traditionally made segmentation and recognition extremely challenging for classical OCR methods. Research consistently shows that this characteristic, which CAPTCHA leverages, is the primary hurdle for machine-based deciphering.
- **Deep Learning Advancements in Recognition:** Despite the complexity, current research is achieving high-accuracy recognition rates for Modi Lipi, primarily through sophisticated **Deep Learning (DL) models**, such as **Convolutional Neural Networks (CNNs)**, **Vision-Language Models (VLMs)**, and **Transfer Learning (TL)** techniques (e.g., using models like AlexNet, VGG16, and ResNet).
    - **High Accuracy Rates:** Several studies report remarkable accuracy rates (often exceeding 95% and in some cases, even near 99%) for both handwritten and printed Modi character recognition, demonstrating that deep learning *can* learn the intricate patterns of the script.
    - **Novel Frameworks for Transliteration:** Notably, a recent framework, **MoScNet** (a VLM developed by IIT Roorkee and collaborators), has been introduced to transliterate Modi script images into Devanagari text, with a focus on historical document preservation. While its initial public-tested success rate on general handwritten documents has been questioned, the fact that a dedicated VLM architecture exists and aims to "crack" the script indicates that the window of security for a Modi Lipi CAPTCHA may be closing.

2. Context of Non-Latin Script CAPTCHA Development

The research landscape for text-based CAPTCHA has shifted away from easily breakable Latin-script designs toward utilising the inherent complexities of other languages and scripts:

- **Focus on Complex Scripts:** There is a clear trend toward developing CAPTCHA based on scripts with complex morphology, such as **Devanagari, Arabic, and Gurmukhi**. These scripts offer features like variable character forms, diacritical marks, and cursive connections that are difficult for generic OCR systems.
- **Need for Robust Obfuscation:** Research in Devanagari and Arabic CAPTCHA frameworks confirms that simply using a non-Latin script is not enough. Success is achieved by applying script-specific **obfuscation techniques**, including:
    - Randomised colour blending (e.g., using Pseudoisochromatic Plates (PIPs)).
    - Overlapping and broken characters.
    - Local and global warping (like arcs and jaws).
    - Using *handwritten* or *pseudo-words* to create variations that are not in a dictionary and cannot be easily segmented.
- **Adversarial Testing:** Newer research emphasises the use of **Adversarial Machine Learning** techniques (like the Jacobian-based Saliency Map Attack - JSMA) to test the security of non-Latin CAPTCHAs, moving beyond basic OCR to actively find and exploit the deep learning models' weaknesses.

3. Motivation:

The core motivation is to capitalize on the features that make Modi-Lipi difficult for machine recognition (current state-of-the-art OCR systems) while still being **solvable by humans** who are familiar with the script, thus serving the dual purpose of security and **script preservation/popularization**.

4. Modi Lipi Script:

The Modi script was the official script used by the Marathas during their reign for writing Marathi. It was used in Maharashtra and throughout many other parts of India. According to [33], the Modi script can be classified into six types: Adyakalin Modi (of the 12th century), Yadavakalin Modi (of the 13th century), Bahmanikalin Modi (of the 14th – 16th century), Shivakalin Modi (of the 17th century), Peshwekalin Modi (of the 18th - 19th century), and Anglakalin Modi (from 1818 till 1952). The Modi script was in use until India's pre-independence era, but later it was superseded by Devanagari.

Importance of Modi script: In medieval India, the Modi script was widely used for scriptures, official documents, and administrative records, covering topics like land, property, yoga, spirituality, and *Ayurvedic* medicine. These fragile records hold valuable insights into medieval history and science. Developing a framework to convert Modi script into recognised scripts would aid transcription, digitisation, and research, benefiting historians and people at large. With 83 million Marathi speakers in India (2011 census [32]), reviving the Modi script could enhance cultural literacy and regional identity.

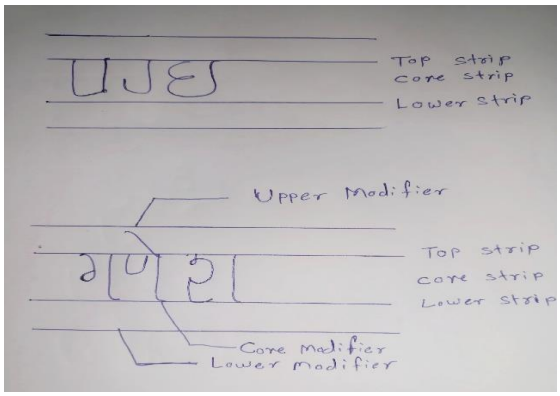


Figure 2

Challenges in transliterating the Modi script: As shown in Figure 3, the Modi script consists of 48 letters, including 36 consonants and 12 vowels. Certain characters have a striking resemblance. The Modi script has some peculiar characteristics that make its transliteration challenging:
(1) First, the *Shirorekha*, a horizontal line equal to the width of the paper, is drawn. Then, a letter is written in a way that it starts and ends at the shirorekha.
(2) Its cursive design was intended for faster writing, resulting in no spaces between words and no clear demarcation of word boundaries.
(3) Ligatures and angular nature.
(4) Very few punctuation marks are observed in the historical texts of the Modi script.

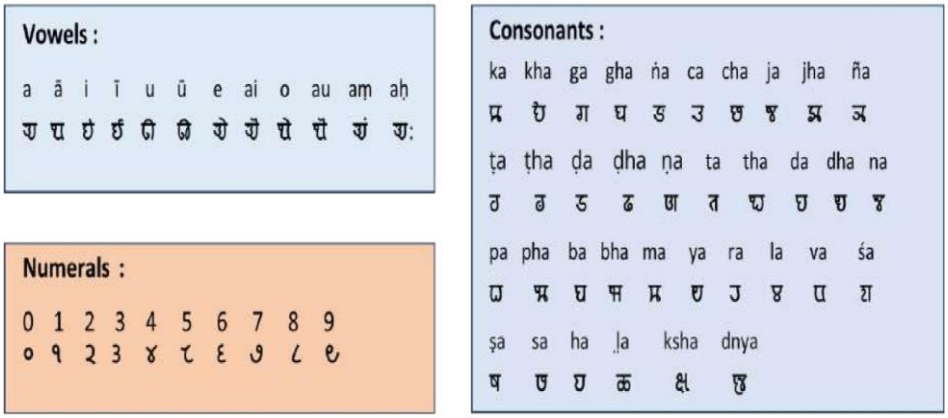


Figure 3: Modi script characters - 12 vowels, 10 numerals, and 36 consonants.

Contributions: Our contributions are as follows:

1. Introducing MoDeTrans Dataset: Our first significant contribution is the **MoDeTrans Dataset**, which comprises 2043 Modi script images and their corresponding Devanagari text transliterations, thereby establishing a valuable benchmark for script transliteration research. Most existing works on Modi Script focus on recognising individual handwritten characters. The datasets used in these studies primarily consisted of particular characters. In contrast, our work is the first to tackle the challenge of *direct transliteration* of handwritten Modi script to Devanagari, as shown in Figure 3. The dataset curation was particularly challenging due to the scarcity of accessible Modi-script documents and the limited number of experts capable of accurately transliterating them into Devanagari. In addition, we have created a synthetic dataset named SynthMoDe Dataset (see Appendix 0.A.1). A contrast between these two datasets highlights the importance of the proposed MoDeTrans dataset, which handles the real-world challenges of Modi script transliteration.
2. Our second major contribution is the vision-language model (VLM) based MoScNet framework for transliterating Modi script images into Devanagari script. MoScNet is inspired by the widespread success of large language models (LLMs) in various downstream tasks. Our key observation is that the transliteration task does not require the computational complexity of full-scale LLMs. Based on this, MoScNet leverages the knowledge distillation (KD) technique, whereby a smaller student model learns transliteration from a "low-rank adaptation" (LoRA) adapted pre-trained teacher model. The use of KD allows MoScNet to remain lightweight and computationally efficient, making it suitable for deployment in low-resource environments.
3. Innovative Transformer Architecture: We propose a new decoder-based transformer design within the student model, featuring *parallel attention* and *QK-normalisation*. To the best of our knowledge, this is the first application of these techniques in a causal modelling setting, significantly enhancing the model's representational capabilities.
4. Superior Performance over Baselines: We conducted comprehensive evaluations spanning diverse architectures—from recurrent neural networks (RNNs) to LLMs. In this analysis, the **MoScNet-XL** configuration, incorporating **429 million parameters** and guided by a **LoRA**-adapted, pre-trained **LlaMA-3 70B** teacher model, demonstrated the best overall performance. This result represents a *notable improvement* compared to all previously evaluated architectures. Similarly, on the optical character recognition (OCR) task, our model performs comparably with the recent models and outperforms previous models.
5. Advancement of Script Transliteration Research: By providing an enriched dataset, a high-performance transliteration model, and a novel transformer-based student architecture, our work lays a strong foundation for future research in script transliteration, resource creation, and related natural language processing tasks. Our framework offers significant utility to historians and researchers. Our technique promotes digitisation and preservation of fragile historical documents, ensuring their accessibility to future generations.

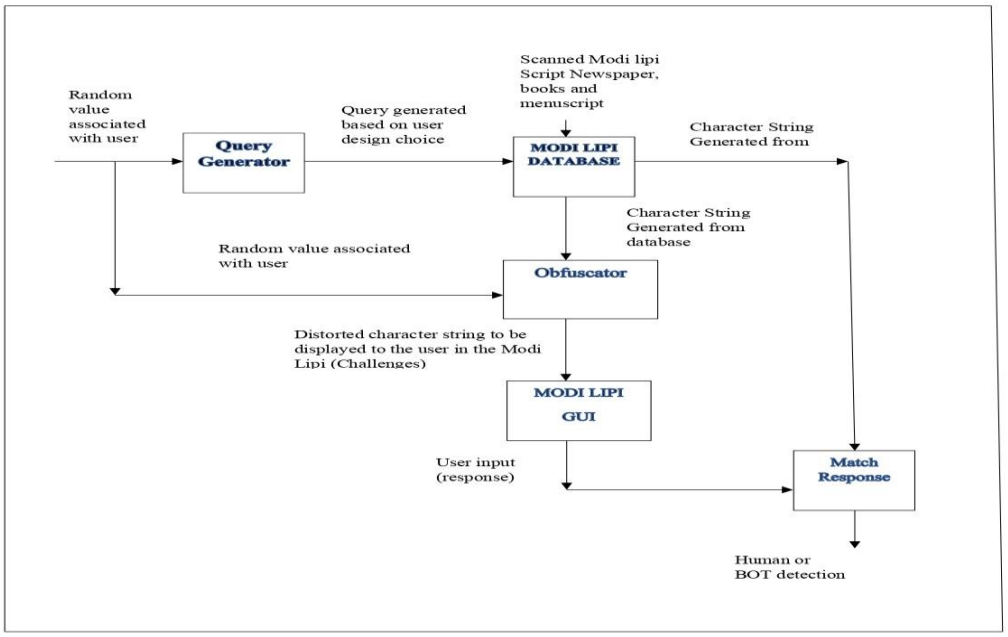
5. Proposed Framework

In this work, we propose a framework (Framework figure) for administering a Modi lipi script-based text CAPTCHA to assist in securing Indian-language web-based applications. The objective of a modilipi CAPTCHA framework is to differentiate a human from a Bot. This can be achieved by testing the

ability of a user to recognize the Modi Lipi characters.

The key components of the proposed modilipi CAPTCHA framework are:

1. MoidlipiDB: A sufficiently large database of modilipi samples (either in text or handwritten form)
2. Query Generator: A mechanism to query the database and obtain a random sample subject to the design.
3. Obfuscator: A module that takes a random sample from the database and distorts noise to it.
4. ModilipiGUI: User challenge response interface.
5. Match Response: A determination of whether the user has submitted an accurate response for the challenge posed.



Modi Lipi CAPTCHA Framework

5.1 ModilipiDB

ModiliopiDB is a collection of books, newspapers, and manuscripts in Modi lipi script. The database is required to be online and can be shared by many web applications. The database can exclusively contain pages in Modi lipi script-based languages or may be multi-lingual, allowing support for many other scripts/languages. The database population process is a one-time major effort with periodic minor updates or additions. The contents are stored as pages, which are uniquely identified as bit objects. The pages are indexed, and a list of words with associated pages is generated. Indexing is also a one-time major effort, with re-indexing performed only on updation. This can be used for randomly selecting a pre-existing word from the database. Similar procedures can be adopted to generate phrases from the pages. In case a random word is required, a sub-module of the modilipiDB module is responsible for the generation of a random word character string with the suitable string length determined by the query generator. The Query Generator module prepares a query that is fed to the Database module, which outputs a random word, a word, or a character string as a Modi lipi script-based character string. Thus, modilipiDB can be used to serve digital content as well as provide the source for generating the challenge. Work is already in progress, partially funded by the Government of India and other agencies, for the creation of Digital Libraries in many research centres and in many Indian languages, not limited to Modilipi script-based languages. These projects aim to digitize Indian language books and manuscripts for better preservation and easy access. In these projects, the initial phase involves scanning old newspapers, manuscripts, and books and storing the content as images. In other projects, the Indian language content is converted into text files using OCR technology. The latter approach is suitable for modilipiDB. Additionally, a web crawler may be developed and used in the Database module that will populate the database with copyright-expired content available on the web. Unicode is

used to support the different types of fonts that are available for the Modi Lipi script.

5.2 Query Generator

The Query Generator module simply generates a query to ModilipiDB that determines whether the text string is a valid word and whether its length is fixed or variable. Other variations can occur if the text string is a phrase or a word. A random number that is associated with the user determines the parameters associated with the text string that is to be generated by the database. The minimum and maximum length of the word, whether we select an existing word from the database or whether we generate a random assortment of characters, are design choices or can be user-specified in this module without compromising the security of the Turing test.

5.3 Obfuscator

The Obfuscator module is the crux of ModiLipi CAPTCHA and uses specific characteristics of Modi Lipi script to deceive the adversary. In the Modi Lipi script, all the individual characters are joined by a head line called "Shiro Rekha". The obfuscator may remove the headline and add noise to the image using patterns like mosaic, arcs/jaws, vertically overlapped on the script. The Obfuscator further misleads the machine by using fonts of varying sizes and character spacing. Sample images that are intended to be used in ModiLipi CAPTCHA are shown below (Figure 4). Segmentation-resistant methods will be employed in its design.

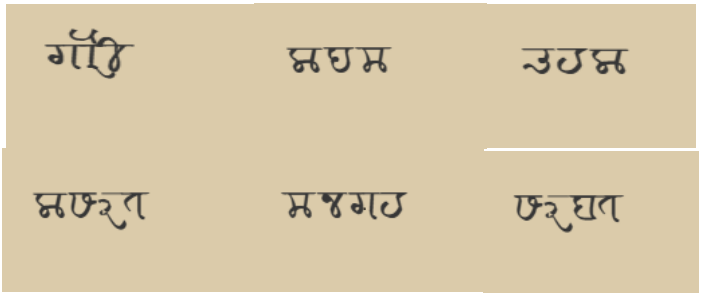


Figure 4

The Obfuscator module operates using a Transformation function that is applied to the character string generated by ModilipiDB. The Transformation function consists of N (in the range of 50-100) parameters, each of which represents a particular type of alteration that can be applied to the character string. The alterations may be at the individual character component level or at the level of the entire character string. Periodically, a subset of m (m < N) parameters is selected randomly that will be used for this time period. Thus, it is ensured that Bots are unaware of which subset of N parameters is currently being used. The random input associated with each user will determine the values of each of the m parameters. For instance, parameters may indicate a font different in word variations. The extent of distortions applied is determined by the degree of robustness and usability that is required. Robustness refers to the ability of a CAPTCHA to prevent Bots from successfully deciphering the character string. Usability refers to easy solvability by humans. A highly robust CAPTCHA successfully deflects Bot attacks but may also present greater difficulty for humans to solve. Thus, robustness and usability aspects have to be carefully balanced. The distortions applied should be difficult for Bots to solve but very easily solved by humans.

5.4 ModiLipiGUI

In ModiLipi CAPTCHA, we display a substantially noisy and distorted image containing the chosen text string or word in the ModiLipiGUI (Figure 5) image display area. A text input box is provided where the user can type in the characters in sequence as they appear in the distorted image. A submit button is to be pressed to signal completion of input.

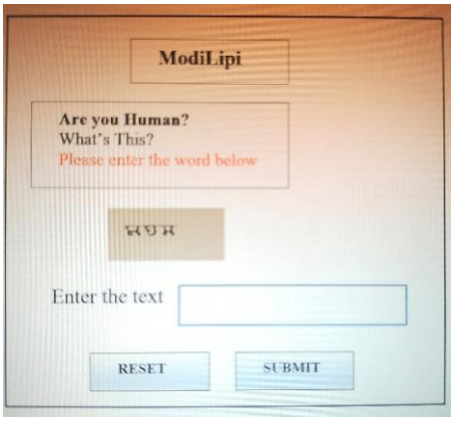


Figure- 5

A timer can be used to cause a CAPTCHA image to expire, giving users a specified amount of time to solve it, while Bots have a limited time to break it. Upon expiration of the current CAPTCHA image, the user can request a new CAPTCHA string by pressing the Refresh button. In case the user is unable to solve the current CAPTCHA, the Refresh button may be used to request a new string. When the user is able to successfully solve the string, he is permitted to move on to the next phase of authentication or request a service from the web application.

5.5 Match Response

The user typed in text from ModiLipiGUI should match the pre-obfuscated string generated by the Database in response to the ModiLipiDB query. If it does, the user is deemed to be a human and may be permitted to authenticate or request web services.

Conclusion

We have seen that by distinguishing between humans and computers, CAPTCHAs offer protection against automated attacks on systems and applications. The criteria for a text-based CAPTCHA's success are robustness and usability. We have outlined techniques in the Obfuscator module to generate challenges that render ModiLipi CAPTCHA robust, as it is resistant to Bot attacks that employ OCR. ModiLipi CAPTCHA is highly usable as it is easy for humans to respond successfully. Since we are using words from books and newspapers or an assortment of characters that are non-words, it is not difficult for humans to visually perceive these characters despite the
distortions and noise due to their superior visual capabilities and cognitive abilities to make connections with words that they have encountered in some context. Distorted images containing random strings are still easy for humans to read, while computers spend endless time processing information. The implementation of ModiLipi CAPTCHA and participation in OCR testing efforts related to Indian language scripts should be taken up as future research work. Handwriting recognition and testing for Modi Lipi script is another future research activity.

References

[1] Von Ahn, L., Blum, M., Hopper, N. J., & Langford, J. (2003).*CAPTCHA: Using Hard AI Problems for Security.* In **Advances in Cryptology — EUROCRYPT 2003** (pp. 294–311). Springer. — The original CAPTCHA paper introduced the concept.

[2] Bursztein, E., Aigrain, J., Moscicki, A., & Mitchell, J. C. (2011). *The End is Nigh: Generic Solving of Text-based CAPTCHAs.* In **WOOT'11: Proceedings of the 5th USENIX Workshop onOffensive Technologies**. A landmark study showing that manyCAPTCHA can be broken with OCR.

[3] Hernandez-Castro, J. C., &Ribagorda, A. (2010). *Pitfalls in CAPTCHA Design and Implementation: The MathCAPTCHA, aCase Study. Computers & Security*, 29(1), 141–157. — DiscussesWeaknesses in math-basedCAPTCHA.

[4] Bock, K., Meier, M., &Holz, T. (2015). *AutoDOS: AutomaticDenial of Service Attacks against Text-based CAPTCHAs.* In**USENIX Security Symposium**. — Explains how CAPTCHA can They themselves are abused in DoS contexts.

[5] Sivakorn, S., Polakis, I., &Keromytis, A. D. (2016). *I Am Robot: (Deep) Learning to Break Semantic Image CAPTCHA.* In **IEEESymposium on Security and Privacy (SP'16)** (pp. 388–403). — Shows deep learning breaking Google's reCAPTCHA.

[6] Ye, G., Tang, Z., Fang, L., & Zhu, X. (2018). *Yet Another TextCaptcha Solver: A Generative Adversarial Network-Based Approach.* In **CCS 18: Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security** (pp.332–348). — Uses GANs to break CAPTCHA.

[7] Alsuhibany, S. A. (2023). *A Survey on Adversarial Perturbations And Attacks on CAPTCHA. Applied Sciences*, 13(7), 4602.https://doi.org/10.3390/app13074602 — A comprehensive recentSurvey of CAPTCHA attacks.

[8] Wen, Y., Yang, C., & Wang, Y. (2023). *Robust Image-Based CAPTCHA Generation Using Adversarial Attack.*In **Proceedings of SPIE 12509, Real-Time Image Processing and Deep Learning 2023** (p. 125091L). — Discusses adversarially-generatedCAPTCHA to resist bots.

[9] Du, X., Liu, X., Zhou, J., Pun, C.-M., Luo, J. (2025). *Unsourced Adversarial CAPTCHA: A Bi-Phase Adversarial CAPTCHAFramework.*

arXiv:2506.10685. — Proposes a two-stage adversarial CAPTCHA defense framework.

[10] Chandra, J., Manhas, P., Kaur, R., & Sahay, R. (2025). *Aura-CAPTCHA: A Reinforcement Learning and GAN-Enhanced Multi-Modal CAPTCHA System.* arXiv:2508.14976. — Uses reinforcement learning + GANs to improve CAPTCHA resilience.

[11] Mori, G., & Malik, J. (2003). *Recognizing objects in adversarial Environments clutter.*

[12] Rajapakse, R., & Fernando, M. (2019). *CAPTCHA mechanisms And AI threats.*

[13] Kulkarni, R. (2018). *Digital Preservation of Modi Script.* The Unicode Consortium, *"The Unicode Standard, Version 7.0 – Modi Script,"* 2014. [Online]. Available: https://www.unicode.org/charts/PDF/U11600.pdf

[14] Google, *"Noto Sans Modi Font,"* Google Fonts, 2024. [Online]. Available:https://fonts.google.com/noto/specimen/Noto + Sans + Modi

[15] S. Anam and D. Gupta, *"Optical Character Recognition for Modi Script," International Journal of ComputerApplications*, vol. 119, No. 7, pp. 1–5, 2015.

[16] S. Varpe and S. Sakhare, *"Recognition of Handwritten ModiScript Using Convolutional Neural Networks," International Journal of Recent Technology and Engineering*, vol. 8, no. 5,pp. 3165–3169, Jan. 2020.

[17] D. Joseph, A. Bhandare, and R. Kale, *"Deep Learning Based Handwritten Modi Script Recognition," International Research Journal of Engineering and Technology (IRJET)*, vol. 8, no. 6, pp. 4200–4205, 2021.

[18] S. B. Kamble and S. D. Joshi, *"Survey of Handwritten Modi Script Recognition Techniques," International Journal of ComputerScience and Information Security*, vol. 18, no. 6, pp. 12–18, 2020.

[19] X. Wu, H. Singh, and R. Krishnan, *"MoScNet: End-to-End Modi Script Transliteration with Knowledge Distillation," Journal ofDocument Analysis and Recognition*, vol. 28, no. 2, pp. 145–160,2025.

[20] L. von Ahn, M. Blum, N. Hopper, and J. Langford, *"CAPTCHA: Using Hard AI Problems for Security," Advances in Cryptology – EUROCRYPT 2003*, Lecture Notes in Computer Science vol. 2656, pp. 294–311, Springer, 2003.

[21] A. Bursztein, R. Beauxis, H. Paskov, D. Perito, C. Fabry, and J. Mitchell, *"The Failure of Text-Based CAPTCHAs,"* in *Proceedings of the 2011 IEEE Symposium on Security andPrivacy*, pp. 343–358, May 2011.

[22] S. P. Jadhav and R. N. Zanjal, *"CAPTCHA Techniques: A Review," International Journal of Computer Applications*, vol. 175,No. 7, pp. 12–18, Oct. 2017.

[23] Von Ahn, L., Blum, M., Hopper, N. J., & Langford, J. (2003).CAPTCHA: Using hard AI problems for security. *Advances in Cryptology*. Recent papers on OCR in Modi script (MoScNet, IJETT 2024).

[24] Yalamanchili, Sushma &Kameswara Rao, M. (2011). *A Framework for Devanagari Script-based Captcha*. arXiv preprint arXiv: 1109.0132. arXiv Proposes a text-based CAPTCHA using Devanagari script, leveraging the difference in reading proficiency between humans and programs. arXiv

[25] Kumar, M., & Jindal, M. K. (2021). *Benchmarks for Designing aSecure Devanagari CAPTCHA*. SN Computer Science. Volume 2, Article number 45. SpringerLink Evaluates various Devanagari CAPTCHA (20 unique ones) focusing on their resilience to de-noising and segmentationattacks. SpringerLink

[26] Sanjay Pate & Rakesh Ramteke et al. (2023). *DEVANAGARI CAPTCHA DATASET OF 1 million Images: A Challenge Test*. Mendeley Data / IEEE DataPort. Mendeley Data+1 Released a large dataset of Devanagari text CAPTCHA (printed & handwritten characters and digits), useful for bothgeneration and recognition tasks. Mendeley Data+1

[27] Arora, S., Bhattacharjee, D., Nasipuri, M., Basu, D.K. & Kundu, M. (2008) "Combining Multiple Feature Extraction Techniques for Handwritten Devnagari Character Recognition", IEEE Region 10 Colloquium andthe Third ICIIS, IIT Kharagpur, India, Dec 2008.