International Journal of Safety and Security Engineering
Vol. 15, No. 9, September, 2025, pp. 1767-1773
Journal homepage: http://iieta.org/journals/ijsse

Dynamic Mirrored CAPTCHA Design and Security Evaluation

Krishna Chythanya Nagaraju[1*], Salim Amirali Jiwani[2], Kiran Kumar Bejjanki[3], Janaki Vinjamuri[4], Kanchipati C. V. Koushik[5], Ratan Kollabathula[5], Dodda Hanoch Raj Jedidiah[5]

[1] Department of CSE, Gokaraju Rangaraju Institute of Engineering and Technology, Hyderabad 500049, India
[2] Department of CSE (AI&ML), Vaagdevi College of Engineering, Warangal 506005, India
[3] Department of Information Technology, Kakatiya Institute of Technology and Science, Warangal 506015, India
[4] Department of Computer Science and Technology, Jayamukhi Institute of Technology and Sciences, Warangal 506322, India
[5] Department of CSE, Gokaraju Rangaraju Institute of Engineering and Technology, Hyderabad 500049, India

Corresponding Author Email: kcn_be@rediffmail.com

https://doi.org/10.18280/ijsse.150901

Received: 16 June 2025
Revised: 16 August 2025
Accepted: 16 September 2025
Available online: 30 September 2025

Keywords:
CAPTCHA, bot, security, OCR, mirror image, inverted, alphabets

ABSTRACT

Completely Automated Public Turing test to Tell Computers and Humans Apart (CAPTCHA) is generally helpful in endorsing that it is human being interacting with the web services but not any attack by a bot. These are used to differentiate between humans and computers, preventing bots from automated operations such as spamming, scraping content. With the advancements of Artificial Intelligence, CAPTCHAs are facing problems because of its ability to bypass captcha using Optical Character Recognizer (OCR) technology. The primary objective of this research is to propose a novel robust CAPTCHA design that can withstand the attacks of bots. It is accomplished by understanding the loopholes of the current CAPTCHA system. This work suggests ways to improve security by implementing CAPTCHA using dynamically positioned rotating inverted alphabet characters, thereby safeguarding internet services from automated bot attacks. The results shown 93% of times OCR failed to correctly identify all the characters of generated CAPTCHA and around 87.5% of participants successfully recognized the characters in CAPTCHA. The project helps the United Nations Sustainable Development Goals (SDGs) to fulfil Goal 9 (Industry, Innovation, and Infrastructure).

1. INTRODUCTION

CAPTCHA, which stands for Completely Automated Public Turing test to Tell Computers and Humans Apart (CAPTCHA), is a security measure which is widely used on websites to protect against automated bot attacks. Its main role is to distinguish between human users and computer programs, and preventing bots from any automated tasks like spamming, scraping data, or unauthorized access. However, with the advancements in technology and artificial intelligence, CAPTCHAs is less effective comparatively. Bots can now use Optical Character Recognizer (OCR) algorithms to read and solve the characters that are used in CAPTCHAs, which are a threat to internet services which are using captchas as security measures. OCR technology was designed to digitize printed text and perform various other tasks. Over time, it has become highly sophisticated and capable of recognizing even the distorted characters in CAPTCHA tests. This advancement means that bots, with the help of artificial intelligence can bypass CAPTCHA, which leads to decreasing their effectiveness as a security measure. This is a serious risk to the integrity of web portals that use CAPTCHAs to protect their services from automated bot attacks. This work addresses the issue by exploring new methods to make CAPTCHA defenses much stronger which will eventually make it difficult for the OCR algorithm to bypass the captcha. Our proposed method is by using dynamic motion CAPTCHAs and mirror image CAPTCHAs, similar to how letters are written in reverse on an ambulance. These approaches help to create CAPTCHA designs that are comparatively more difficult, if not impossible, for OCR algorithms to bypass. By implementing these CAPTCHA techniques, authors hope to enhance the security of web services and make it much harder for bots to bypass these protective measures using OCR technology.

In this work, integration of dynamic motion and mirror image CAPTCHA is done. Dynamic motion CAPTCHAs involve characters or objects that move in unpredictable patterns, making it difficult for OCR algorithms to capture and interpret them. Mirror image CAPTCHAs display characters in reverse, adding another layer of complexity for OCR systems. These innovative CAPTCHA designs aim to significantly enhance security by presenting challenges that are more resistant to automated recognition by bots.

Further the work is divided in to five sections with first section being Introduction, second section takes care of Literature survey of related work of this field and Proposed Model is discussed in section three, whereas Section four concentrates on Implementation and Results. The paper ends with a fifth section containing Conclusion and future work possibilities.

2. LITERATURE SURVEY

A thorough survey of the literature in the specific field of the project was made. After going through numerous research papers and articles, a list of the 20 most relevant papers was shortlisted, and the crux of the work done by other authors was summarized as given below. Chang et al. [1] underscored the importance of developing CAPTCHAs that are both secure and user-friendly, offering valuable insights and recommendations for future improvements in this area. The study suggested usability concerns for CAPTCHAs when blindly enhancing their robustness. Deep learning techniques and neural style transfers were used in developing CAPTCHAs and were deployed with Taobao, GeeTest, NetEase, Tencent, and Vaptcha. The study mentioned that Tencent achieved 89.80% accuracy, whereas Taobao CAPTCHA achieved 100% accuracy, and GeeTest, NetEase, and Vaptcha scored 91%, 88%, and 87.5% respectively. The research by Dinh and Hoang [2] summarized the use of image processing, machine learning (ML) algorithms, computer vision (CV), supervised learning, Discrete Fourier Transform (DFT), Hidden Markov Models (HMMs), the Regularized Least-Squares Classification (RLSC) algorithm, and Amazon Mechanical Turk for labeling data in developing CAPTCHAs. But the work lacked specific quantitative data on the performance of different CAPTCHA schemes. Hayashi et al. [3] explored a Study on Robustness Evaluation of Weave Filters against Reverse Image Search Attacks on CAPTCHA, where processed images were made to withstand search engine attacks. The original images were first prepared, followed by cropping, resizing, and applying a weave filter. However, images with subjects that resemble the processed shape or are close-up might still be easily recognized. The average accuracy was recorded as 85% in their work. *zxCAPTCHA: New Security-Enhanced CAPTCHA* was the title of the research conducted by Dinh et al. [4] in which the authors studied cognitive-based, image-based, and text-based CAPTCHA characteristics using TensorFlow and Torch as deep learning frameworks and explored VGG-16, ResNet-101, ResNet-50, and LeNet-5 neural networks for classification. The researchers made use of the EMNIST dataset for text and the ImageNet ILSVRC-2012 dataset for images. Implementing the proposed zxCAPTCHA required a significant number of computational resources, making it highly complex. Zhao et al. [5] explored a two-stage framework for breaking text-based CAPTCHAs called GEESOLVER, leveraging Transformer-based models. Transformer models are computationally complex, and still a lot of research is going on to study the dependencies of it. The accuracy percentages for different samples were recorded as 92.9% for Yandex and 97.8% using Wikipedia. Raut et al. [6] proposed a robust CAPTCHA scheme for web security, but it had vulnerability to advanced techniques of OCR, highlighted concerns regarding resource intensiveness and complexity for website integration, and exhibited an accuracy of almost 98% when used by humans. This work had shortcomings like complexity for website integration and a high potential for overfitting. Sachdev [7] demonstrated the potential of deep learning techniques but underscored the need for further refinement to improve accuracy and mitigate misclassification issues. The dataset used consisted of 1069 text-based CAPTCHA images, which were available on Kaggle. The work was experimented on using multi-task learning CNN (MTL-CNN) and SVM. The manual labor requirement in SVM-based recognition, limited accuracy, susceptibility to misclassification of characters, and misinterpretation in CAPTCHA images owing to noisy lines in the image were certain shortcomings of this work. Using MTL-CNN they showed an accuracy of 98% and 92% when SVM was used.

Object recognition, segmentation attacks, deep learning, low-cost attacks, Recurrent Networks, GAN-based solvers, chaos-based CAPTCHAs, adversarial examples, and convolutional neural networks were techniques deployed by de Cerqueira et al. [8]. Bias in data selection, static analysis, limited generalizability, incomplete coverage, and assumptions about CAPTCHA effectiveness were certain issues that needed further work, though they exhibited an accuracy of 99.8% using RNN.

Ardhita and Maulidevi [9] employed techniques like handwritten digits from the MNIST dataset, real-world object images, or images from the ImageNet dataset to generate CAPTCHAs. The authors utilized a Generative Adversarial Network model trained on the MNIST digits for CAPTCHA generation. The adversarial examples were tested on a pre-trained ILSVRC Xception model. The model failed to correctly classify all 10 generated images. This may inadvertently have become too difficult for human users to solve, resulting in poor user experience. Combining alphanumeric characters with distorted backgrounds, text distortion, image labeling, and pattern recognition resulted in an increase in the computational burden on users' devices, as observed by Baolin and Minhuan [10]. Ensemble Adversarial Training, Differentiable Approximation for Image Preprocessing, and stochastic Image Transformation techniques were employed by Zhang et al. [11] for generating robust CAPTCHAs. This resulted in computational overhead in ensemble training, reliance on transferability between models, and challenges in maintaining effectiveness under diverse stochastic transformations. It was observed that only 83.0% accuracy was achieved with a high difficulty level. Bi and Liu [12] generated a synthetic dataset using a Python script. The dataset consisted of 3755 popularly used Chinese characters, out of which 80 images per character were used for training and 20 images per character were used for testing purposes. The authors employed Convolutional Neural Networks, an n-gram model, and object detection using Faster RNN. While the proposed system performed well for Chinese character CAPTCHAs, its generalizability to other languages or CAPTCHA types was limited. They demonstrated an accuracy of 99.21% in Chinese character recognition. Weng et al. [13] made use of a synthetic image dataset developed using images from ImageNet, Baidu.com, 12306.com, and synthetic character generation. They used automated web crawlers to gather CAPTCHA-solving service sites, image recognition algorithms to solve CAPTCHAs, and performed a market analysis of CAPTCHA-solving strategies. They observed faster solving times by automated CAPTCHA solvers with a success rate of up to 0.85. Ince et al. [14] matched with the current author's idea to some extent, where the researchers made use of PHP (Personal Home Page) programming language for implementation of a CAPTCHA algorithm involving splitting CAPTCHA images into parts with random rotation values. The algorithm's performance was evaluated with a compilation and run-time duration of 1.12 to 1.17 seconds per session and accuracy on a localhost Apache Server. However, some patterns may be challenging for older or disabled human users. Mittal et al. [15] tried to break the image CAPTCHAs, and the researchers made use of the

Facebook CAPTCHA dataset besides the ImageNet dataset. Applying the Convolution Neural Network Inception V3 model techniques with transfer learning, they achieved a best prediction class, such as Guitar and Lion, of 100% and 98% respectively. They successfully handled Facebook images but the model was unable to map certain CAPTCHA images to labels in the ImageNet dataset.

A good study on the applications of deep learning architectures such as CNN and YOLO v3 DCNN, CaffeNet, SE-ResNeXt-50, and R-CNN in building and breaking CAPTCHAs can be observed in the article [16]. Ding et al. [17] observed that no specific systematic work was carried out earlier to evaluate the Multimodal Large Language Models (LLM) impact on different kinds of CAPTCHAs. They proposed the Illusion CAPTCHA. The researchers conducted a user study to check how many attempts a user took to successfully pass the CAPTCHA generated by them by applying Zero-Shot and Chain of Thought prompting. It was mentioned that 86.95% of testers successfully cleared the CAPTCHA test. The work revealed that LLMs and humans made similar mistakes in identifying the characters of CAPTCHA quite often. A set of 4 research questions was addressed by the authors to evaluate the effectiveness of the work done and they identified the majority of users' perception with a confidence metric of 4.8. Work done by Derea et al. [18] made use of a refined visual system for CAPTCHA recognition using an adapted UPDown image captioning model. The researchers made use of four datasets including Captcha 0.3, Weibo, BoC, and Gregwar and also used a dual layer long short-term memory structure with a better attention mechanism to decode the CAPTCHA. They used both top-down and bottom-up attention mechanisms for building a Recurrent Neural Network architecture. An accuracy of 96.89% in recognizing CAPTCHA was registered for BoC CAPTCHAs. The refined visual attention (RVA) used, based on the context of the already generated words of language on the go, adjusted its weights for visual attention. The CNN model they used for feature extraction consisted of 5 convolution and 5 max pooling layers with ReLU activation function. They included one extra layer for the regular visual attention layer which preceded the sigmoid function. The scope of improvement was observed when the CRRVA model used on Gregwar CAPTCHA could only yield 47.08% accuracy in recognizing characters of the CAPTCHA correctly. Whereas, Kumar et al. [19] proposed a Mirror CAPTCHA that was built using mirrored fonts of numbers and characters combined with distortion and complex backgrounds. They opined that the existing image-based algorithms would need more byte space, which caused an increase in the bandwidth cost for a particular page. Ray et al. [20] focused on the usage of adapted NST (Neural Style Transfer) in designing image-based CAPTCHA called a Style Matching CAPTCHA. It was found to be effective and recorded an accuracy of 95.61% by participants in solving it. They recorded an accuracy of only 37% in breaking the developed CAPTCHA using ResNet-50 and Inception-v3.

Keeping in view of all the above works, this research tried to balance security with usability by employing an intuitive CAPTCHA that included both normal and mirrored characters with dynamic motion, ensuring it remained user-friendly while being difficult for automated systems to crack. The breadth and depth of the literature were evaluated to study the different kinds of CAPTCHA generation and how vulnerable those CAPTCHAs were. It was evident from the literature studied that there was a need for a more resilient CAPTCHA generation mechanism to combat attacks of bots. This study differed from existing work in the way CAPTCHA was generated, as almost all studies used only simple alphabets in CAPTCHA but not mirror-imaged ones, and also in dynamic positioning with its circular motion.

3. PROPOSED MODEL

To address the challenges posed by bots circumventing security measures through OCR technology, the proposal is to develop a secured web service with implementation of an advanced CAPTCHA system. OCR technology, while sophisticated, enables bots to bypass traditional CAPTCHA images and gain access to web services. This poses a significant threat to the security and integrity of online platforms, which makes it necessary implementation of more robust countermeasures. The proposed advanced CAPTCHA system aims to raise the barrier for unauthorized access attempts by implementing innovative and dynamic CAPTCHA mechanisms. These mechanisms will include complex image recognition tasks to effectively differentiate between human users and automated bots. The CAPTCHA system will be adaptive, continuously evolving to counteract new and emerging OCR techniques employed by bots. Implementing such an advanced CAPTCHA system will enhance the security of our web service, ensuring that legitimate users can access the platform seamlessly while preventing unauthorized and malicious access attempts. The advanced CAPTCHA system will provide a robust defense mechanism, making it exceedingly difficult for bots to bypass the security measures and breach the system. It was concluded that following are the main objectives of research.

3.1 Increase the complexity of CAPTCHA

To enhance security against automated attacks, this approach includes introducing intricate visual patterns, dynamic elements, and inverted characters into proposed CAPTCHA system. By increasing the complexity of these CAPTCHA challenges, it is aimed to significantly hinder OCR algorithms used by bots to bypass text. Dynamic elements such as moving objects or changing patterns further complicate the task for automated systems, ensuring that only human users can successfully interpret and complete the CAPTCHA. Distorted characters, which involve warping, rotating, or fragmenting the text, add an additional layer of difficulty, making it harder for OCR algorithms to accurately recognize and decode the characters.

3.2 Optimize the design of the CAPTCHA

While increasing the complexity of CAPTCHA challenges is crucial, it is equally important to ensure that these challenges remain user-friendly for legitimate human users. To achieve this, work will fine-tune the visual elements of the CAPTCHA to maintain text legibility while simultaneously challenging OCR algorithms. This involves careful balancing of distortion levels and the intricacy of visual patterns so that human users can easily interpret and solve the CAPTCHA without experiencing frustration. By optimizing the design, it is ensured that the CAPTCHA is accessible and comprehensible to genuine users, thereby enhancing the user experience and

reducing the likelihood of abandonment due to difficulty.

3.3 Conduct OCR-based testing to check the robustness of the novel captcha

To ensure the effectiveness and robustness of generated novel CAPTCHA system, OCR-based testing using static image of generated CAPTCHA is to be conducted. This involves simulating attack scenarios where OCR algorithms attempt to recognize and decode the CAPTCHA challenges. By analyzing these attack attempts, one can quantify the success rates of OCR recognition and identify potential weaknesses in CAPTCHA design. This iterative testing process allows refining and enhancing the CAPTCHA system continuously. By systematically evaluating and improving the CAPTCHA based on these test results, ensures that security measures remain resilient against sophisticated automated attacks, providing a reliable defense mechanism for our web service.

With the reference to the system architecture as shown in Figure 1, project is divided into the following modules: User Authentication Module, Login Page Module, Captcha Module, and Registration Page Module. The User Authentication Module is designed to handle both user login and registration processes. The login functionality allows registered users to securely access the application by entering their username or email address along with their password. This ensures that only authorized users can gain entry to the application. The registration functionality enables new users to create an account by providing a username, email address, password, and confirmation of the password.

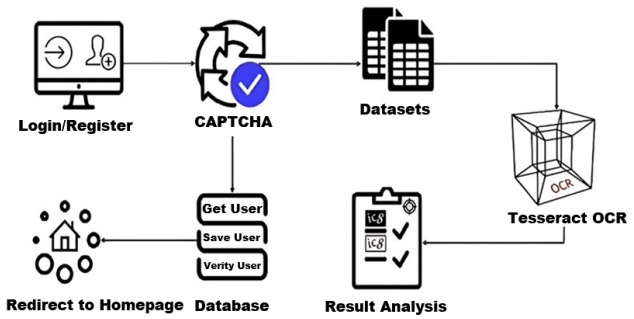

Figure 1. Proposed system's architecture

This process ensures that all necessary user details are collected and verified before account creation. The Login Page Module is responsible for presenting the login interface to the user. It displays the login page with input fields where users can enter their username or email and password. To enhance user experience and security, this module includes a feature that allows users to toggle the visibility of their password. By clicking on an eye icon, users can choose to either reveal or conceal their password as they type, reducing the likelihood of input errors. The Captcha Module is integrated to add an extra layer of security to the login and registration processes. It generates a unique CAPTCHA image each time the page is loaded, incorporating mirror alphabets and motion effects to make it challenging for automated scripts to decipher. This module also provides a refresh option, allowing users to generate a new CAPTCHA image if the current one is difficult to read.

This ensures that the CAPTCHA remains user-friendly while maintaining its effectiveness in preventing automated logins. Registration Page Module is designed to facilitate the account creation process for new users. It displays the registration page with input fields for a username, email address, password, and password confirmation. This module ensures that all required information is collected in a structured manner. Additionally, it performs validations to confirm that the passwords match, enhancing the accuracy and security of the registration process. Together, these modules work seamlessly to provide a secure and user-friendly authentication system for the application, ensuring that only authorized users can access the system while preventing unauthorized access through automated scripts. The Captcha Evaluation Module is designed to assess the robustness and effectiveness of the CAPTCHA system integrated within the application. This module generates multiple CAPTCHA images and subjects them to OCR based testing, simulating potential automated attacks. By analyzing the success rates of these OCR attempts, the module helps identify any weaknesses in the CAPTCHA design. This iterative testing process involves generating CAPTCHA images, applying OCR tools, and analyzing the results to continuously refine and enhance the CAPTCHA system. This ensures that the CAPTCHA remains resilient against sophisticated automated attacks, providing a reliable defense mechanism for the application's authentication system. In situations of low band width at the use end the proposed model may induce some delay and unrest to the users.

4. IMPLEMENTATION AND RESULTS

For this work, two strings are defined: one contains normal alphabets, and the other contains mirrored and inverted alphabets. A charMap object is created to map mirrored characters to their normal counterparts. This mapping is useful for validation purposes. An array of fonts contains seven different font styles: Roboto, Open Sans, Courier Prime, Pacifico, Lobster, Raleway, and Merriweather. These styles will be applied randomly to each character to make the CAPTCHA harder for bots to recognize. In general, most CAPTCHAs have 5 to 6 characters, but in this CAPTCHA model, we confine it to 4 characters to prevent it from becoming overly complex. This CAPTCHA includes 3 mirrored or inverted alphabet characters and 1 normal alphabet character. A random index (normalCharIndex) within the CAPTCHA length is chosen to determine which character will be a normal alphabet character, with the rest being mirrored characters. To randomize the position of the CAPTCHA inside the CAPTCHA container so as to avoid possible threat by bot when using static captcha, an array of cases containing different top and left positions is defined. This adds a layer of difficulty for bots. This array contains 9 cases of top and left position. On refreshing the captcha, a new CAPTCHA will be generated randomly in those positions. Since this is a dynamic motion CAPTCHA, to reduce its complexity, the user can enter the CAPTCHA characters in any order to be considered correct. Once the user successfully registers, they will be redirected to the login page. After successfully logging in, they will be redirected to the main page. The CAPTCHA generated in this work can be seen in Figure 2 and Figure 3. The CAPTCHA is not clear as it is a dynamic rotation-based CAPTCHA with letters rotating in cylindrical shape. The algorithmic representation of the logic with pseudo code is presented in the Algorithm AnimatedMirroredCaptcha.

Algorithm: AnimatedMirroredCaptcha

Purpose:
//To generate a CAPTCHA containing one normal and 3 mirrored characters and to provide //the ability of dynamic motion to the characters of the CAPTCHA and dynamic positioning //of the CAPTCHA in the CAPTCHA container.

Input: Null

Output: Animated Captcha containing mirrored characters and with dynamic positioning is generated inside the CAPTCHA container.

Pseudo code:

ALGORITHM AnimatedMirroredCaptcha
// Variables:
// alphabet ← "acefghijkorstvxyzABCDEFGHJKLNOQRSTUVXYZ"
// mirrored ← "ɐɔpǝɟƃɥᴉɾʞlɯuodbɹsʇnʌxʎz"
// charMap ← mapping of mirrored → normal characters (for later validation)
// chars[4] ← the four '' elements in the DOM
// container ← the '.captcha-image' element (bounding box for chars)
// wrapper ← the '.captcha-container' element (parent of '.captcha-image')
// correctChars[4] ← array to store exactly what was drawn
// positions[] ← list of predefined {top,left} cases for relocating the wrapper
// positions[] ← [{ top: '-40%', left: '0%' }, { top: '-40%', left: '60%' }, { top: '130%', left: '60%' },{ top: '130%', left: '0%' },{ top: '90%', left: '0%' }, { top: '-40%', left: '30%' },{ top: '90%', left: '60%' }, { top: '90%', left: '30%'}, {top: '130%', left: '30%'}]

PROCEDURE generateCaptcha()
Step 1: Choose which slot will hold the one "real" character
SET normalIndex ← RANDOM_INT(0, 3)

Step 2: For each of the 4 slots, pick & display a char, position it, and set its animation delay
FOR *i* FROM 0 TO 3 DO
Step 2a: Select the source pool
IF *i* = normalIndex THEN
SET pool ← alphabet
ELSE
SET pool ← mirrored
END IF

Step 2b: Draw a random character
SET c ← pool.charAt(RANDOM_INT(0, pool.length − 1))
DISPLAY_CHAR_AT_SLOT(*i*, *c*)
correctChars[*i*] ← *c*
Step 2c: Compute a random position inside the container
SET maxX ← container.width − chars[*i*].width
SET maxY ← container.height − chars[*i*].height
SET x ← RANDOM_FLOAT(0, maxX)
SET y ← RANDOM_FLOAT(0, maxY)
SET chars[*i*].style.left ← *x* + "px"
SET chars[*i*].style.top ← *y* + "px"
Step 2d: Stagger the float animation so chars move out of sync
SET chars[*i*].style.animationDelay ← (*i* × 0.5) + "s"
END FOR
Step 3: AFTER all characters are placed, position the entire CAPTCHA block
// Pick one of the predefined positions and apply it to the wrapper
SET p ← positions [RANDOM_INT(0, positions.length − 1)]
SET wrapper.style.top ← p.top
SET wrapper.style.left ← p.left
END PROCEDURE
// Event Bindings:
ON document.DOMContentLoaded DO
CALL generateCaptcha()
END

To ensure CAPTCHA robustness, OCR-based testing using pytesseract and OpenCV was used. CAPTCHA images are preprocessed—converted to grayscale, noise reduced with Gaussian Blur, deskewed, and enhanced with adaptive thresholding and morphological operations. Tesseract then performs OCR on the images. Recognized text is compared to expected text, considering mirrored and inverted character mappings. By analyzing the accuracy of character recognition, we identify and address potential weaknesses in the CAPTCHA design, ensuring resilience against automated attacks. This iterative testing process helps maintain the CAPTCHA's effectiveness.

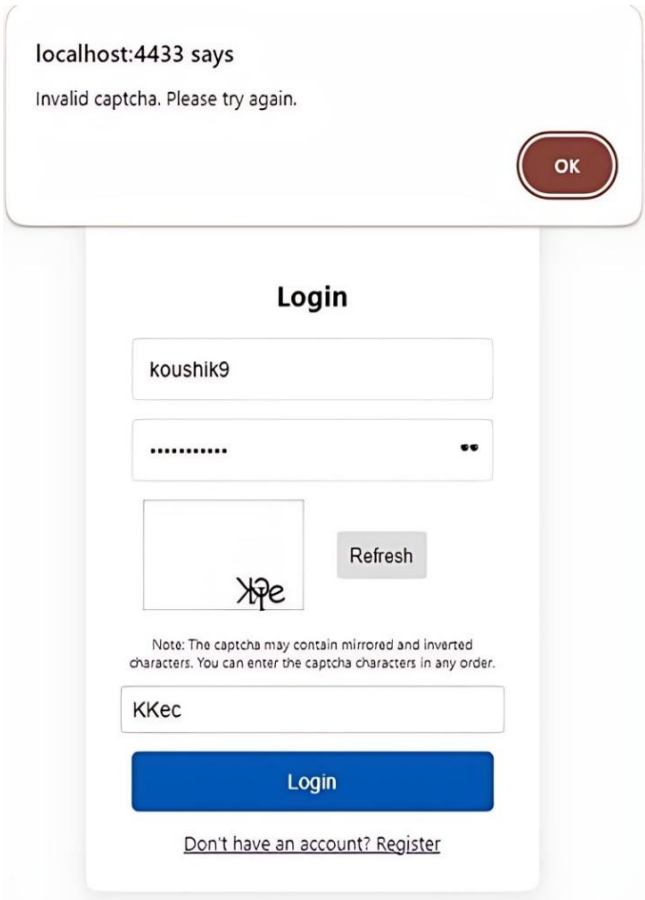

Figure 2. The screenshot of application developed showing message for wrong CAPTCHA entry

For the evaluation of our CAPTCHA model, we performed OCR on PNG images in a specified directory using Tesseract via the pytesseract library, with preprocessing handled by OpenCV to enhance recognition accuracy. It starts by configuring the Tesseract executable path and defining a dictionary expected_texts mapping image filenames to their expected text. The script includes a mirrored_inverted_map for handling mirrored or inverted characters. Each image is processed through the preprocess_image function, which reads the image, converts it to grayscale, applies Gaussian blur to reduce noise, and deskews it by correcting any rotation. Adaptive thresholding converts the image to binary, and morphological operations remove small noise. The main function, perform_ocr, iterates over each PNG image in the directory, preprocesses it, and performs OCR with a custom configuration (--oem 3 --psm 6). The recognized text is cleaned and compared to the expected text using the compare_texts function, which counts correct and incorrect characters by comparing character frequencies.

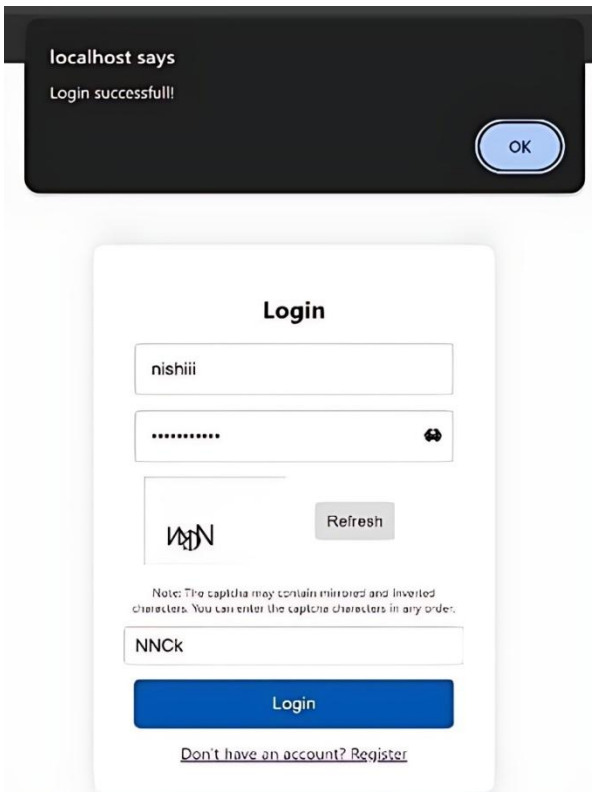

Figure 3. The screenshot of application developed showing login successfully on entering correct CAPTCHA

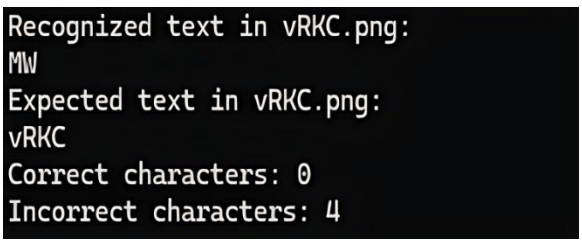

Figure 4. The output of OCR test conducted for generated novel captcha of this research

Results for each image, including recognized text, expected text, correct and incorrect character counts, are printed. All recognized characters across images are collected, sorted alphabetically, and printed at the end. The script includes error handling to catch and print any issues during file reading or processing. This process ensures that images are optimally prepared for OCR, maximizing recognition accuracy and providing a detailed comparison of OCR results against expected values. A sample result for OCR test conducted is shown in Figure 4. It was observed that the OCR test failed to identify correct characters 93% of time it was applied. However, the users could successfully recognize the characters of CAPTCHA with an accuracy of up to 87.5%.

5. CONCLUSION

In conclusion, the rapid advancements in artificial intelligence and OCR technology present significant challenges to the effectiveness of traditional CAPTCHA systems. This research underscores the vulnerability of these systems to sophisticated OCR-based attacks, highlighting the urgent need for more robust and innovative CAPTCHA designs. The proposed advanced CAPTCHA mechanisms, such as dynamic motion along with mirror imaged alphabets based CAPTCHAs, offer promising solutions to enhance web service security. These designs challenge the capabilities of OCR technology, making it significantly harder for bots to bypass security measures. Additionally, focus on optimizing CAPTCHA designs ensured that while security is heightened, the challenges remain user-friendly for legitimate human users, thus balancing security with user experience. The experimental results showed that the CAPTCHA developed is more resilient to attacks as compare to existing captchas thus meeting the three objectives considered. This, in turn, supports the United Nations Sustainable Development Goal 9 by fostering innovation and building resilient infrastructure. The results shown 93% of times OCR failed to correctly identify all the characters of generated CAPTCHA and around 87.5% of participants successfully recognized the characters in CAPTCHA. Ultimately, by staying ahead of evolving bot technologies, the developed advanced CAPTCHA systems offer a robust defense mechanism, safeguarding online platforms and ensuring a secure digital environment for businesses and consumers alike. The future of research on advanced CAPTCHA systems to counter OCR-based attacks is promising. Further work can focus on integrating AI and machine learning to create adaptive, resilient CAPTCHAs with a goal to optimize user experience by balancing security with usability i.e. Adaptive CAPTCHA design integrated with AI can be explored.

REFERENCES

[1] Chang, G., Gao, H., Pei, G., Luo, S., et al. (2024). The robustness of behavior-verification-based slider CAPTCHAs. Journal of Information Security and Applications, 81: 103711. https://doi.org/10.1016/j.jisa.2024.103711

[2] Dinh, N.T., Hoang, V.T. (2023). Recent advances of Captcha security analysis: A short literature review. Procedia Computer Science, 218: 2550-2562. https://doi.org/10.1016/j.procs.2023.01.229

[3] Hayashi, N., Satoh, T., Uehara, S. (2023). A study on robustness evaluation of weave filters against reverse image search attacks on CAPTCHA. In 2023

International Conference on Consumer Electronics-Taiwan (ICCE-Taiwan), PingTung, Taiwan, pp. 533-534. https://doi.org/10.1109/ICCE-Taiwan58799.2023.10226760

[4] Dinh, N., Nguyen, T., Truong, V. (2023). zxCAPTCHA: New security-enhanced CAPTCHA. In 2023 15th International Conference on Knowledge and Smart Technology (KST), Phuket, Thailand, pp. 1-6. https://doi.org/10.1109/KST57286.2023.10086931

[5] Zhao, R., Deng, X., Wang, Y., Yan, Z., et al. (2023). GeeSolver: A generic, efficient, and effortless solver with self-supervised learning for breaking text captchas. In 2023 IEEE Symposium on Security and Privacy (SP), San Francisco, USA, pp. 1649-1666. https://doi.org/10.1109/SP46215.2023.10179379

[6] Raut, Y., Pote, S., Boricha, H., Gunjgur, P. (2022). A robust captcha scheme for web security. In 2022 6th International Conference on Computing, Communication, Control and Automation (ICCUBEA), Pune, India, pp. 1-6. https://doi.org/10.1109/ICCUBEA54992.2022.10011113

[7] Sachdev, S. (2020). Breaking captcha characters using multi-task learning CNN and SVM. In 2020 4th International Conference on Computational Intelligence and Networks (CINE), Kolkata, India, pp. 1-6. https://doi.org/10.1109/CINE48825.2020.234400

[8] de Cerqueira, J.A.S., de Almeida, P.S., Canedo, E.D., de Oliveira Alves, G., Giozza, W.F., de Mendonça, F.L.L., de Sousa, R.T. (2020). Exploratory overview on breaking CAPTCHAs using the theory of the consolidated meta-analytic approach. In 2020 15th Iberian Conference on Information Systems and Technologies (CISTI), Seville, Spain, pp. 1-6. https://doi.org/10.23919/CISTI49556.2020.9140983

[9] Ardhita, N.B., Maulidevi, N.U. (2020). Robust adversarial example as captcha generator. In 2020 7th International Conference on Advance Informatics: Concepts, Theory and Applications (ICAICTA), Tokoname, Japan, pp. 1-4. https://doi.org/10.1109/ICAICTA49861.2020.9429048

[10] Baolin, X., Minhuan, Z. (2020). A solution of text based CAPTCHA without network flow consumption. In 2020 IEEE 11th International Conference on Software Engineering and Service Science (ICSESS), Beijing, China, pp. 395-399. https://doi.org/10.1109/ICSESS49938.2020.9237694

[11] Zhang, J., Sang, J., Xu, K., Wu, S., et al. (2020). Robust CAPTCHAs towards malicious OCR. IEEE Transactions on Multimedia, 23: 2575-2587. https://doi.org/10.1109/TMM.2020.3013376

[12] Bi, X., Liu, X. (2020). Chinese character captcha sequential selection system based on convolutional neural network. In 2020 International Conference on Computer Vision, Image and Deep Learning (CVIDL), Chongqing, China, pp. 554-559. https://doi.org/10.1109/CVIDL51233.2020.00-28

[13] Weng, H., Zhao, B., Ji, S., Chen, J., Wang, T., He, Q., Beyah, R. (2019). Towards understanding the security of modern image captchas and underground captcha-solving services. Big Data Mining and Analytics, 2(2): 118-144. https://doi.org/10.26599/BDMA.2019.9020001

[14] Ince, I. F., Yengin, I., Salman, Y.B., Cho, H.G., Yang, T.C. (2008). Designing CAPTCHA algorithm: Splitting and rotating the images against OCRs. In 2008 Third International Conference on Convergence and Hybrid Information Technology, Busan, Korea (South), pp. 596-601. https://doi.org/10.1109/ICCIT.2008.195

[15] Mittal, S., Kaushik, P., Hashmi, S., Kumar, K. (2018). Robust real time breaking of image captchas using inception v3 model. In 2018 Eleventh International Conference on Contemporary Computing (IC3), Noida, India, pp. 1-5. https://doi.org/10.1109/IC3.2018.8530607

[16] Moradi, M., Moradi, M., Palazzo, S., Rundo, F., Spampinato, C. (2024). Image CAPTCHAs: When deep learning breaks the mold. IEEE Access, 12: 112211-112231. https://doi.org/10.1109/ACCESS.2024.3442976

[17] Ding, Z., Deng, G., Liu, Y., Ding, J., Chen, J., Sui, Y., Li, Y. (2025). IllusionCAPTCHA: A CAPTCHA based on visual illusion. In Proceedings of the ACM on Web Conference 2025, Sydney, Australia, pp. 3683-3691. https://doi.org/10.1145/3696410.3714726

[18] Derea, Z., Zou, B., Kui, X., Abdullah, M., Thobhani, A., Abdussalam, A. (2025). A novel CAPTCHA recognition system based on refined visual attention. Computers, Materials & Continua, 83(1): 115. https://doi.org/10.32604/cmc.2025.062729

[19] Kumar, A., Lavanya, Sundar, S. (2014). Captcha using mirror fonts for security against OCR. https://api.semanticscholar.org/CorpusID:12058106.

[20] Ray, P., Bera, A., Giri, D., Bhattacharjee, D. (2023). Style matching CAPTCHA: Match neural transferred styles to thwart intelligent attacks. Multimedia Systems, 29(4): 1865-1895. https://doi.org/10.1007/s00530-023-01075-0