IJIRAE:: International Journal of Innovative Research in Advanced Engineering ISSN: 2349-2163
Volume 11, Issue 12, December 2024 https://www.ijirae.com/archives

Automatic Captcha Recognition Technology Based on AI

Utkarsh Kumar Singh
Department of Information Technology,
Greater Noida Institute of Technology, (Engg. Institute),
Greater Noida, INDIA
ishu2611kumar@gmail.com
Vikrant Bharti
Department of Information Technology,
Greater Noida Institute of Technology, (Engg. Institute),
Greater Noida, INDIA
bhartivicky2002@gmail.com
Yash Raj
Department of Information Technology,
Greater Noida Institute of Technology, (Engg. Institute),
Greater Noida, INDIA
yashktiwari2003@gmal.com
Vishal Verma
Department of Information Technology,
Greater Noida Institute of Technology, (Engg. Institute),
Greater Noida, INDIA
vks567899@gmail.com
Pawan Mishra
Department of Information Technology,
Greater Noida Institute of Technology, (Engg. Institute),
Greater Noida, INDIA
pawan.it@gniot.net.in
Prof.Dr.Vikas Singhal
Professor & Head, Department of Information Technology,
Greater Noida Institute of Technology, (Engg. Institute),
Greater Noida, INDIA
profvikassinghal@gmail.com
Dr.Ajay Sahu
Professor, Department of Information Technology,
Greater Noida Institute of Technology, (Engg. Institute),
Greater Noida, INDIA
ajaysahu.it@gniot.net.in

Publication History
Manuscript Reference No: IJIRAE/RS/Vol.11/Issue12/DCAE10080
Research Article | Open Access | Double-Blind Peer-Reviewed | Article ID: IJIRAE/RS/Vol.11/Issue12/DCAE10080
Received:10,November 2024,Revised: 21,November 2024,Accepted: 30, November 2024, Published Online: 13,December 2024. https://www.ijirae.com/volumes/Vol11/iss-12/01.DCAE10080.pdf
Article Citation: Utkarsh, Vikrant, Yash, Vishal, Pawan, Prof.(Dr.)Vikas, Dr.Ajay (2024),Automatic Captcha Recognition Technology Based on AI. IJIRAE:: International Journal of Innovative Research in Advanced Engineering, Volume 11, Issue 11 of 2024 pages 834-841
Doi:> https://doi.org/10.26562/ijirae.2024.v1112.01
BibTeX Key: Utkarsh@2024Automatic

Abstract: Captcha (Completely Automated Public Turing test to tell Computers and Humans Apart) is doing an phenomenal critical object for safeguarding online systems from automated bots and ensuring user authentication. With advancements in Artificial Intelligence (AI), captcha recognition technology has undergone significant transformation, allowing automated systems to decipher even highly complex captcha formats with remarkable accuracy. This paper dives onto application of AI methodologies, with addition of Convolutional Neural Networks (CNNs) for effective captcha solving. Through the use of various advanced algorithms, AI systems excel in tasks such as text recognition, segmentation, and noise reduction, addressing traditional challenges like distortion and visual clutter. The study also evaluates the associated security risks, ethical considerations, and potential counterstrategies to strengthen system defenses.

Keywords: Captcha, Artificial Intelligence, Machine Learning, Convolutional Neural Networks, Security, Automation

I. INTRODUCTION

CAPTCHA, an acronym for "Completely Automated Public Turing Test to Tell Computers and Humans Apart," is a widely worked security mechanism designed to differentiate humans with automated programs. In the beginning of early 21st century, CAPTCHAs has evolved into a standard defense against malicious bots and denial-of-service (DoS) attacks. These tests are intentionally crafted to be solvable by humans but challenging for automated systems, making them a critical tool for internet security. CAPTCHA has many varsities various forms, consists of various jumbled alphabets, visually visible img, and voice-based tests, and are extensively used to protect online services. While researching the topic of CAPTCHA recognition plays a vital role in identifying vulnerabilities and improving future designs, helping developers enhance their robustness against increasingly sophisticated automated threats. Text-based CAPTCHAs remain a widely used and effective security method to prevent automated attacking because of simplicity and broad applicability. These CAPTCHAs typically consisting of capital letters and lowercase letters and numeric values (0–9), though more extensive alphabets union, such as China based attack malwar high risk characters, have been introduced in recent years. To enhance their security, text-based CAPTCHAs employ various mechanisms, including unwanted auditorial disturbences, also blurred images of written texts character rotation, wraping, differential computable letters lengths, also covering grouping up of alphabetical words may that be in capital form or small form.

On the other hand if we take a look at, traditional methods for solving CAPTCHAs, which rely on character segmentation followed by recognition, struggle with newer designs where characters are skewed and cannot be easily separated by vertical lines. This challenge renders rectangular segmentation methods ineffective and calls for more advanced classification techniques. [12]Solution proposed by God fellow, leverages in depth Convolutional Neutral Networks (CNNs) an powerful framework also applied in object classification, automatic speech recognition, and natural language processing. Despite these advancements, the rapid evolution of deep learning has made CAPTCHA recognition systems increasingly adept at bypassing many current text-based CAPTCHA defenses, necessitating continuous innovation in CAPTCHA design.

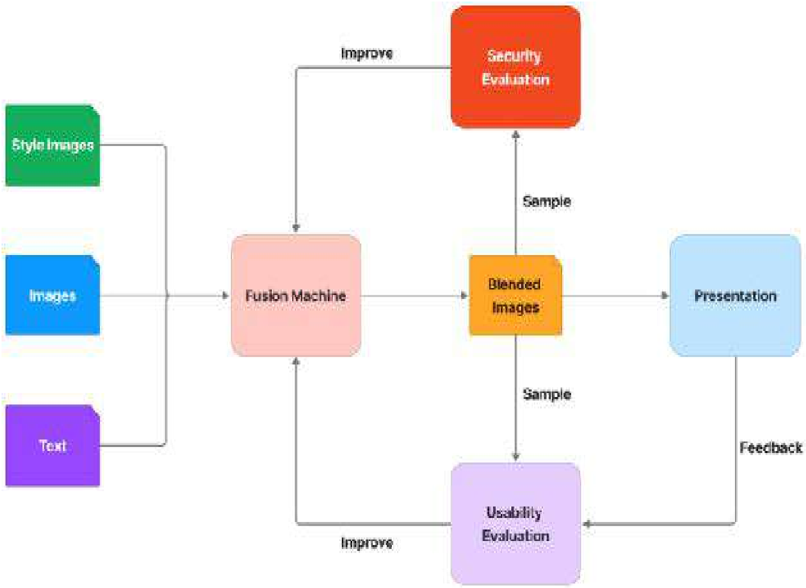

Figure1: Architecture of CAPTCHA recognition system

II. LITERATURE REVIEW

A dominant security methods in the beginning era of 21st century. However, various threatening of leveraging analysis of visual pics, identifying any possible relation between them also containg also of advanced level were developed for compromise these systems. To counteract such vulnerabilities, introduced to enhance their security. Despite these efforts, advancements in artificial intelligence (AI) eventually enabled the resolution of distorted text CAPTCHAs with near-perfect accuracy, leading to a decline in the use of character included functions. As text recognition challenges became easier for AI systems, suffers like image classification and recognition emerged as more complex alternatives. This shift gave rise to image-, involving pull and place tasks, picking of visal pics, mechanisms, designed to differentiate humans from bots. However, between 2013 and 2018, advanced CV and ML solutions successfully bypassed many prominent image-based CAPTCHA systems, including widely used variants.[2] Techniques such as distortion, background noise mixing, and enhance security against ML-based attacks. Despite these advancements, incremental learning approaches have been developed to overcome even adversarial CAPTCHA's. To accommodate visually impaired users, researchers also introduced audio-based CAPTCHAs, further diversifying the range of CAPTCHA methods. The application of image recognition to create CAPTCHAs marked a significant innovation, leading to the development of various image-based CAPTCHA systems. These are generally categorized onto 3 parts: pics selection CAPTCHAs, rat-pulling CAPTCHAs, also taking CAPTCHAs.

Pictured picking up of captchas is a necessary need for the identify and take out specific pics a given explanation, with several popular implementations widely used across platforms. Neural networks have been the most effective tools for cracking these CAPTCHAs, showcasing the capability of modern machine learning techniques. [11]

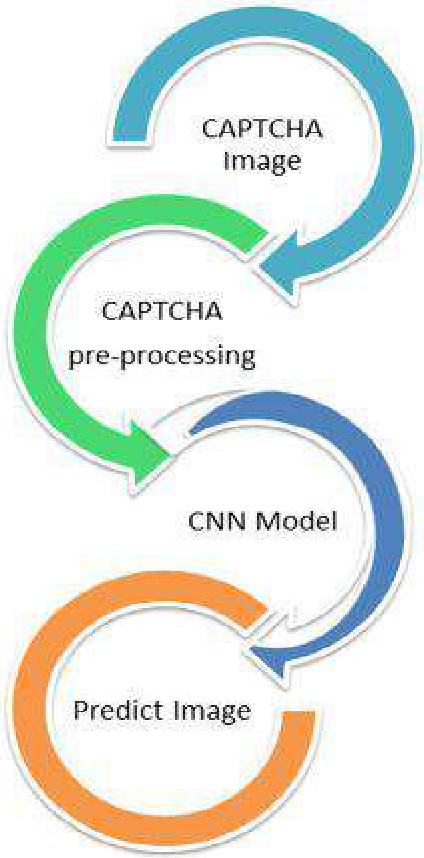

Figure 2: Flowchart of generation of captcha

Mouse-dragging CAPTCHAs involve tasks like dragging an image fragment for its original location or rotating an image to the correct orientation. Although these CAPTCHAs add an interactive element, they have been defeated through low-cost automated approaches, highlighting persistent security vulnerabilities. To address these weaknesses, researchers have explored the use of adversarial examples, which introduce subtle alterations to CAPTCHA designs, making them more robust against automated attacks. Innovations like Deep CAPTCHA have also emerged, integrating immutable adversarial noise to counter deep-learning-based attacks effectively. Additionally, advanced methods, such as those leveraging generative adversarial networks (GANs) with transfer learning, have demonstrated the potential to compromise even sophisticated CAPTCHA systems, underscoring the ongoing need for novel and adaptive security mechanisms.

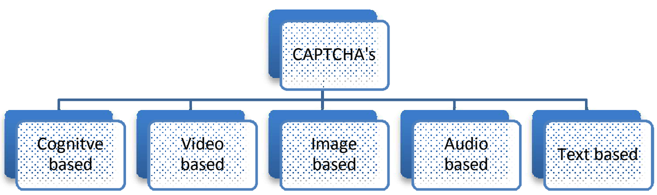

Figure 3: Main schemes of CAPTCHA

To overcome the limitations of insufficient CAPTCHA division architecture, researcher had turned to algos capable of recognizing CAPTCHAs not having dividing. These segmentation-free simultaneously identify all CAPTCHA characters without breaking them into individual segments. By assigning specific neurons for provided part, the description into various parts process is streamlined. The following approach offers high recognition speed and eliminates the need for segmentation. However, as the number of CAPTCHA characters increases, the model's storage size also grows enormous as result of explanation. Our methodology is aimed at recognizing CAPTCHAs having jointed, mix matched alphabets. Their library enhances recognition accuracy by considering correlations between adjacent characters.

However, it uses distinct also full jointed parts of every possible, leading to a more complex architecture and increased storage requirements. [9] Similarly, introduced a written doc CAPTCHA recognition new method for CNNs and focal loss. While their method addresses challenges in CAPTCHA recognition, it has limitations, including the lack of resulted in comparison of techniques and the absence of practically testing validate its practical correctness.

III. METHODOLOGY

Recent studies using deep learning have achieved significant success in solving CAPTCHAs. However, conventional CNN approaches often lose critical features during transitions between convolution and pooling layers. To overcome this limitation, the proposed study integrates skip connections and adopts a 5-fold validation strategy to reduce bias. The CAPTCHA recognition framework, shown in Figure 1, follows a structured process. Data is first normalized through image processing techniques to enhance compatibility with data divided into individual characters, enabling the development of an OCR-like model that accurately identifies characters. The 5-fold validation results demonstrate strong performance.[8] The proposed modules is a architecture. It contains of three key components: t images to produce binary image copies. The CNN extracts features from these images, while the softmax layers classify the characters. Provides details of the ABI model, outlines the complete model architecture.

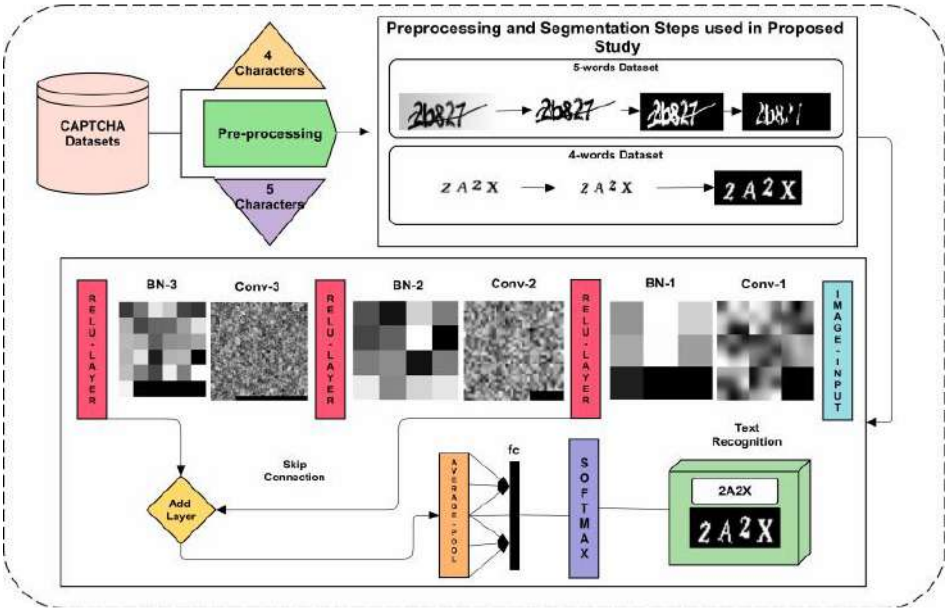

Figure 4: Flow chart of emotion detection algorithm

An (ABI) designed to help locate characters within a group. Each ABI consists of with a vertical white bar that shifts position across the image. For the first ABI, the white bar starts on the left side and moves incrementally to the right in subsequent ABIs until it reaches the right edge.[6] ABIs are used to divide the CAPTCHA's kk characters into nn groups. The first ABI identifies the first group of characters, the second ABI identifies the second group, and so forth. Within each group, a corresponding softmax layer recognizes the characters. For example, the first softmax layer detects the first character, the second softmax layer detects the second character, and so on is combined with a copy of the original CAPTCHA image to create an Attached Binary Image with CAPTCHA Copy (ABICC). This process generates nn ABICC images, each responsible for locating and recognizing a specific group of characters. These ABICCs are sequentially fed into the CNN, where the ABIs provide group location information, and the softmax layers identify characters within each group. Through this method, the CNN is trained to locate and recognize all characters in the CAPTCHA image efficiently.

3.1 Dataset

The give study utilizes Kaggle, each containing CAPTCHA images with either five or four characters. The five-character dataset (d1) comprises 1204 images, while the 4charactered (d2) includes 8459images. The datasets feature a mix of numeric and alphabetic characters, with 19 character types in d1 and 32 character types in d2. Provides details on the dimensions and file extensions of the datasets before and after segmentation, while illustrates the character frequency distribution for both datasets. The frequency and number of characters differ between the two datasets. The d2 dataset includes more characters and higher frequencies, although it lacks complex inner line intersections and text merging. In contrast, the d1 dataset has fewer characters and lower frequencies but contains more complex data. Initially, d1 images have dimensions of 40 × 100 × 2, where 51 representing row, 150 represents columns2 indicates color depth. The d2 images are smaller, with dimensions of 24 × 72 × 3. Both datasets share similar character locations, allowing for manual cropping to isolate individual characters for model training. However, since the dimensions of segmented characters may vary, resizing is necessary to standardize them. The put pics for are in Portable Network Graphics (PNG) format and did not require format conversion. After segmentation, each character is resized to 20 × 24 pixels, ensuring adequate representation of their visual binary patterns.

Table 1. Few recent CAPTCHA recognition-based studies methods and their results

Reference	Year	Dataset
Wang & Shi (2021)	2021	CNKI CAPTCHA, Random Generated, Zhengfang CAPTCHA
Ahmed &Anand (2021)	2021	Tamil, Hindi and Bengali
Bostik et al. (2021)	2021	Private created Dataset
Kumar & Singh (2021)	2021	Private
Dankwa& Yang (2021)	2021	4-words Kaggle Dataset
Wang et al. (2021)	2021	Private GAN based dataset
Thobhani et al. (2020)	2020	Weibo, Gregwar

Details of the datasets before and after resizing are summarized. The images in the d1 dataset do not require extensive image processing for segmentation and normalization. In contrast, the d2 dataset necessitates additional steps to eliminate the central intersecting line between characters, ensuring accurate character isolation. For the d1 dataset, three preprocessing steps are applied: converting the images to grayscale, transforming them into binary format, and then taking their complement. In the d2 dataset, two additional steps—erosion and area-based selection—are performed to remove the intersecting lines and refine the character edges.[10]

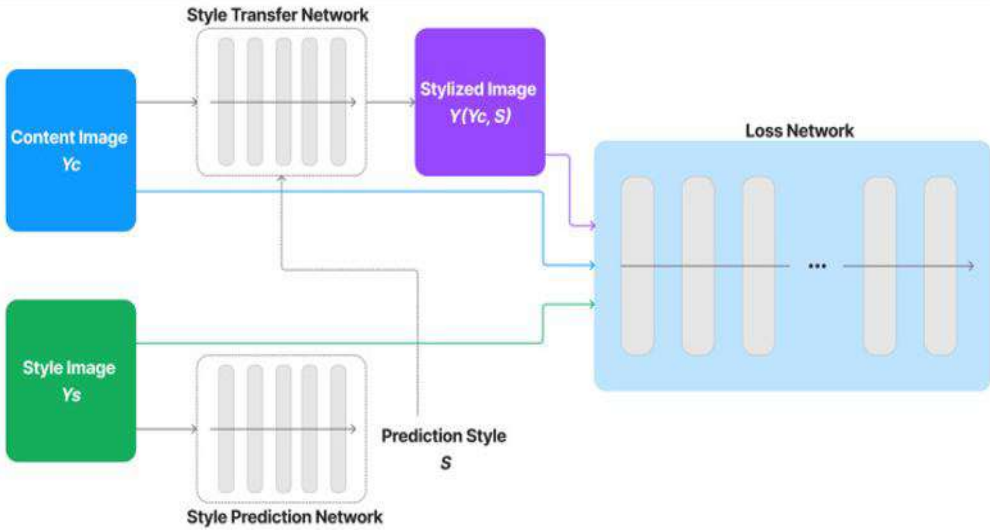

Figure5: Overall architecture of the system.

The proposed study addresses limitations in traditional CNN-based methods, which often lose critical information when pooling feature maps from previous layers. To mitigate this, a one-skip connection is integrated while maintaining standard convolutional blocks. Additionally, inspired by the K-Fold validation method, the study splits both datasets into five folds for training and testing. The results from these folds are averaged to report the final accuracy.[3] The proposed CNN model comprises 16 layers, organized into three main blocks, each containing convolutional, batch normalization, and ReLU layers. After these nine layers, a skip connection integrates inputs from all three blocks. This is followed by average pooling, fully connected, and softmax layers to complete the architecture.

Properties of a good Captcha

When implementing a new CAPTCHA design, it should first be deployed on a test site without additional security measures to focus solely on evaluating its effectiveness against automated attacks. [7] The testing period should be sufficiently long to gather meaningful research insights.

To enhance security against bots, the CAPTCHA should incorporate the following features:

- Randomness and Uniform Distribution: All parameters should exhibit randomness with a uniform distribution. For text CAPTCHAs, this includes aspects such as the number and arrangement of areas, lines, and pixels with varying properties (e.g., color, grouping, group size), a variable number of characters, diverse typefaces, and flexible image sizes.
- Consistent Difficulty: All CAPTCHA variants, such as visual and audio challenges, should maintain the same level of difficulty to avoid simpler alternatives.
- Alignment with AI Challenges: The design should closely mimic the original AI problem it aims to test.
- Bypass and Relay Attack Mitigation: The CAPTCHA should include mechanisms to detect and counteract automated bypass methods or relay attacks.
- Uniform Challenge Distribution: Challenges should be distributed uniformly, independent of users and their responses. Similarly, answers should be random and uniformly distributed, with no statistical correlation between challenges and their solutions.
- Bot Detection Complexity: Employ techniques such as adversarial samples, unique response mechanisms, or specialized communication methods with CAPTCHA servers to make it harder for bots to ascertain the correctness of their responses.

The proposed study leverages the K-fold validation technique, splitting the data into five subsets to mitigate bias in both training and testing phases. This method ensures that no data is left out of the testing process and that all data contributes to training, with the final results representing the mean of all models. As a result, the outcomes are more reliable compared to traditional CNN approaches. The d2 dataset benefits from clear, structured segmentation in its images, while the d1 dataset features less distinct isolated text images, leading to lower classification performance for d1. In contrast, the d2 dataset yields higher and more reliable classification results, making it more effective as a CAPTCHA solver. The performance for each character and dataset in each fold is discussed in the following section.[1]

Data Preprocessing: The dataset consists of images of size 50 height and 200 width. These images first need to be pre-processed before developing the model. For the purpose of training, the images in the dataset have a filename same as the CAPTCHA possessed by the image. The images are first pre-processed by reading them in grayscale. This helps us to remove the noise to some extent as the images get invariant of the background color. Simultaneously, the file name is also stored in a string. Each grayscale image is then scaled and reshaped. Then we create an array of dimension 5*36, used to store the character present at each position in the CAPTCHA. At each position, through filename we find which characters are present at each position of the CAPTCHA and update the corresponding location to one in this array. This array will thus be used for training the model. Thus, after pre-processing we obtain all the grayscale images and the target array containing information regarding the characters present in each CAPTCHA image.

Model Development

The model developed for this CAPTCHA dataset uses CNN. It consists of a total twenty four layers comprising the input layer, convolutional layers, max pooling layers, dense layers, flatten layers and dropout layers. The total numbers of parameters is 1,818,196 where 1,818,132 parameters are trainable and 64 are non-trainable parameters. A brief architecture of the layers is depicted in figure. The CAPTCHA prevents attacks by frequently altering its signal attributes, such as the colors used to indicate correct or incorrect image selections, every five seconds. It also changes the positions and contents of unselected image cells after any image is picked. To bypass this, an automated bot must repeatedly simulate the process of clicking the hint symbol to retrieve information and relay it to a remote user. The user then resolves the challenge multiple times. This process requires a complex communication mechanism between the bot and the remote user, resulting in a resolving time that exceeds the CAPTCHA's time limit.

The CAPTCHA is designed in a way consists of key components, including a fusion mechanism, a security evaluation system, and a usability evaluation process, as depicted in Figure 2. The fusion mechanism utilizes techniques such as seamless cloning, neural style transfer, adversarial examples, and defense methods to integrate text, images, and random style elements into a cohesive design. A subset of these blended images undergoes a security evaluation to measure the CAPTCHA's resilience against automated attacks. This assessment incorporates advanced deep neural network (DNN) attacks designed to analyze and classify objects within the CAPTCHA. Useful information extracted from the security evaluation the process is transferred to the fusion mechanism to enhance the fusion process and bolster its resistance to attacks. Additionally, certain blended samples or feedback challenges are directed to the usability evaluation process to assess the CAPTCHA's user-friendliness, such as determining whether users can easily recognize and solve it or identifying the difficulties users encounter. A group of testers participates in the usability evaluation, tasked with resolving challenge samples and offering recommendations to improve the CAPTCHA's usability. These insights are then translated into refined designs within the fusion mechanism, further enhancing the CAPTCHA's user experience. The available 0 and 1 values are inverted at each pixel position ax and by. This inverted image serves as an isolation step for the d2 dataset. For the d1 dataset, an additional erosion process is applied. Erosion is a morphological operation that uses a structuring element defined by its shape to modify the binary image. This structuring element removes pixels from the image based on its geometry. In the case of CAPTCHA images, intersecting lines are removed using a line-shaped structuring element. This element operates on a neighbourhood of pixels, and for the study, a line of length 5 with a 90-degree orientation is used. This approach effectively removes intersecting lines from each character in the binary image, as demonstrated in row 1. The erosion process is specifically tailored to a length of 5 and an angle of 90 degrees.

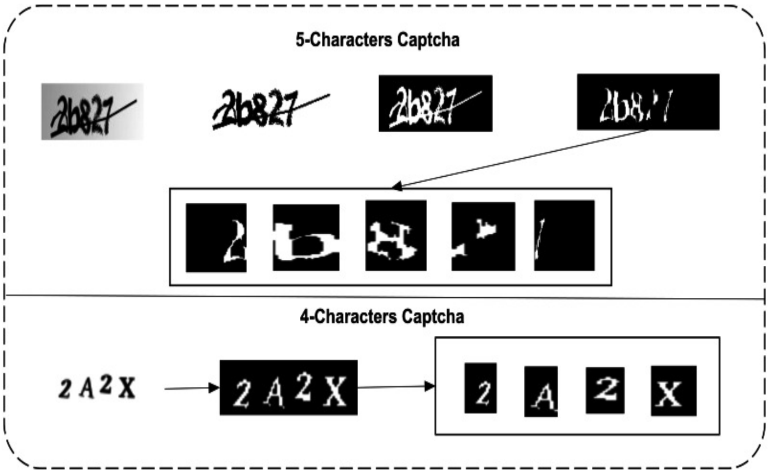

Figure 6: 4 and 5 characters CAPTCHA

IJIRAE:: International Journal of Innovative Research in Advanced Engineering ISSN: 2349-2163
Volume 11, Issue 12, December 2024 https://www.ijirae.com/archives

The system's effectiveness is measured through indicators like accuracy, precision, recall, and F1-score. Its resilience is examined by testing the model on CAPTCHAs with differing levels of complexity. Based on the assessment outcomes, the model is refined to mitigate any limitations, such as adapting to novel distortion methods or accommodating larger datasets. While doing the research we found out that, our automatically CAPTCHA recognition system depend on the previous stage of process, advanced machine learning techniques, and iterative refinement to attain superior accuracy and performance. Contemporary systems increasingly utilize CNNs, RNNs, and Transformers to address the escalating intricacy of CAPTCHA designs while preserving speed and dependability.

IV. RESULT AND DISCUSSIONS

There are 19 types of characters with varying fold-by-fold accuracy. The mean accuracy across all folds is summarized in the last row of the table. Among the characters, *Y* has the highest validation accuracy (94.31%), likely due to its distinct structure compared to others. The second highest accuracy is achieved by *G* (94.07%), which is almost equal to *Y*. These two characters exceed 95% recognition accuracy, while other characters range between 79% and 88%. The character *M* has the lowest accuracy at 61.98%, varying significantly across folds (54% to 73%). This suggests that *M* often resembles other characters, requiring structural refinement to improve its recognition accuracy. To mitigate CAPTCHA vulnerabilities and enhance user accessibility, characters with higher performance may require added structural intricacy, such as angular deviations, to withstand machine learning algorithms. Refining the CNN architecture, including modifications to skip connections, could further improve accuracy. The lower accuracy observed in this dataset may be attributed to a smaller training set and a limited number of images. While other studies on comparable datasets report higher accuracies, they may lack the robustness offered by the unbiased validation approach employed in this analysis. This section presents the precision, recall, and F1-Score across all folds. The character Y achieves the highest F1-Score at 96.15%, followed closely by G at 94.88%. The overall average F1-Score across all folds is 86.32%, slightly surpassing the overall accuracy. Since the F1-Score balances precision and recall, it offers a more dependable metric, especially for imbalanced classes, contributing to the robustness of the proposed method. Results for the four-character dataset are detailed in the following section.

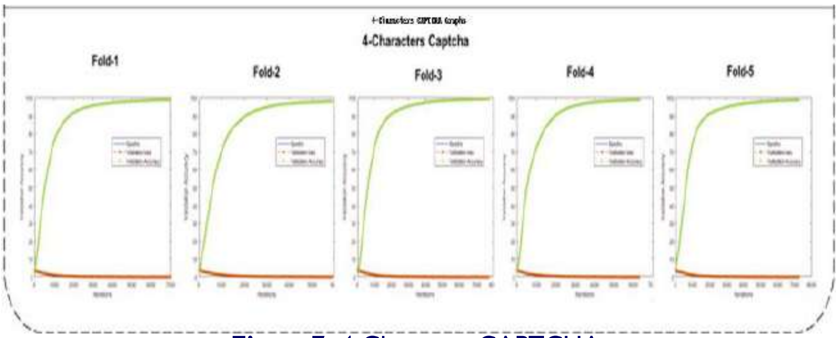

Figure7: 4 Character CAPTCHA

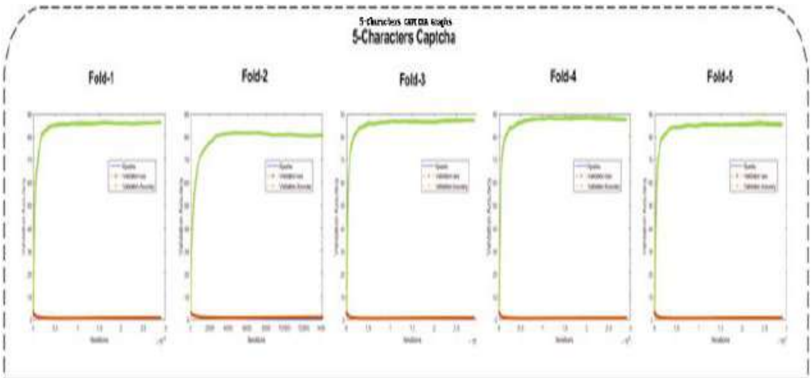

Figure 8: 5 Character CAPTCHA

These frameworks are designed to streamline the resolution of CAPTCHA challenges, frequently utilizing machine learning models such as Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) to manage the intricacies of text, image, or combined CAPTCHA types. By preprocessing information to filter out disturbances and refine inputs, the framework improves identification precision. It also undergoes repetitive training and evaluation to adjust to diverse CAPTCHA formats and alteration methods. Although these frameworks are mainly employed in studies to assess CAPTCHA resilience, they emphasize the importance of crafting tasks that harmonize user-friendliness for individuals with robustness against automated systems. To bypass this, an automated bot must repeatedly simulate the process of clicking the hint symbol to retrieve information and relay it to a remote user. The user then resolves the challenge multiple times. We propose generating a sufficiently extensive dataset of challenge-response pairs (C=challenge, R=response C = \text{challenge}, R = \text{response}) for CAPTCHA evaluation.

To detect irregularities in the data distribution, we utilize randomness analysis and statistical methods, focusing on:

- Inconsistencies in RR: These might signal vulnerabilities to generic or blind attacks.
- Inconsistencies in CC: These could uncover patterns that aid in classifying challenge types and analyzing their structural design.
- Correlations between CC and RR: These may reveal exploitable weaknesses in side-channel attacks.

These analyses can involve evaluating basic properties of CC, such as color histograms, region sizes, pixel value distributions, distances between similar areas, maximum and minimum values within data segments, or bit correlations with specific patterns. These techniques assist in estimating a CAPTCHA system's security parameters, facilitating the detection of vulnerabilities, such as configuration settings that unintentionally disclose sensitive information.

V. CONCLUSION

The research presents an innovative method for deep learning to tackle CAPTCHA challenges. It employs a skip-connected CNN architecture to efficiently decode text-based CAPTCHAs. Two CAPTCHA datasets are thoroughly examined and assessed on a character-by-character basis. To ensure impartial outcomes, the research applies a five-fold cross-validation technique, boosting confidence in the results. The findings reveal notable advancements compared to earlier studies, emphasizing the susceptibility of these CAPTCHA formats. The study concludes that such designs are highly vulnerable to exploitation by malicious actors. The proposed CNN model showcases its capability to evaluate CAPTCHA designs in real-time with enhanced dependability. Furthermore, leveraging publicly accessible datasets for training and evaluation strengthens the method's reliability and reliability in addressing text-based CAPTCHAs. Deep learning techniques have been widely studied in CAPTCHA-solving research, highlighting the importance of CAPTCHA designs that reduce user frustration while effectively countering automated solvers. Enhancing CAPTCHA security is vital for protecting online systems from malicious threats. In the future, methods like data augmentation and advanced data generation could be employed to develop more robust CAPTCHA datasets. For instance, CAPTCHAs featuring intersecting lines, which are harder to solve, could be further investigated. Additionally, deep learning models could be adapted to handle CAPTCHAs in multiple local languages, expanding their usability. Furthermore, our results endorse the notion of synergy, indicating that a compound hierarchical model, leveraging the strengths of individual standalone models, may offer the most efficient approach.

REFERENCES

1. Inayat U., Zia M.F., Mahmood S., Khalid H.M., Benbouzid M. Learning-Based Methods for Cyber Attacks Detection in IoT Systems: A Survey on Methods, Analysis, and Future Prospects. Electronics. 2022;11:1502. https://doi:10.3390/electronics11091502 [DOI] [Google Scholar]
2. von Ahn L., Blum M., Hopper N.J., Langford J. Eurocrypt. Springer; Berlin/Heidelberg, Germany: 2003. CAPTCHA: Using Hard AI Problems for Security. [Google Scholar]
3. Szegedy C., Zaremba W., Sutskever I., Bruna J., Erhan D., Goodfellow I.J., Fergus R. Intriguing properties of neural networks. arXiv. 20131312.6199 [Google Scholar]
4. Gatys L., Ecker A., Bethge M. A Neural Algorithm of Artistic Style. arXiv. 2015 https://doi:10.1167/16.12.326.1508.06576 [DOI] [Google Scholar]
5. Gao H., Tang M., Liu Y., Zhang P., Liu X. Research on the Security of Microsoft's Two-Layer Captcha. IEEE Trans. Inf. Secur. 2017;12:1671–1685. https://doi:10.1109/TIFS.2017.2682704. [DOI] [Google Scholar]
6. Gao H., Yan J., Cao F., Zhang Z., Lei L., Tang M., Zhang P., Zhou X., Wang X., Li J. The Network and Distributed System Security Sym-Posium. NDSS; SanDiego, CA, USA: 2016. A Simple Generic Attack on Text Captchas; pp. 1–14. [Google Scholar]
7. Gao H., Wang W., Qi J., Wang X., Liu X., Yan J. The robustness of hollow CAPTCHAs; Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security; Berlin, Germany. 4 November 2013; pp. 1075–1086. [Google Scholar]
8. Bursztein E., Aigrain J., Moscicki A., Mitchell J.C. The end is nigh: Generic solving of text-based CAPTCHAs; Proceedings of the 8th {USENIX} Workshop on Offensive Technologies ({WOOT} 14); San Diego, CA, USA. 19 August 2014. [Google Scholar]
9. Goodfellow I.J., Bulatov Y., Ibarz J., Arnoud S., Shet V. Multi-digit number recognition from street view imagery using deep convolutional neural networks. arXiv. 20131312.6082 [Google Scholar]
10. Chew M., Tygar J.D. the International Conference on Information Security. Springer; Berlin/Heidelberg, Germany: 2004. Image recognition captchas; pp. 268–279. [Google Scholar]
11. Datta R., Li J., Wang J.Z. Imagination: A robust image-based CAPTCHA generation system; Proceedings of the 13th annual ACM International Conference on Multimedia; Singapore. 6–11 November 2005; pp. 331–334. [Google Scholar]
12. Goswami G., Powell B.M., Vatsa M., Singh R., Noore A. FR-CAPTCHA: CAPTCHA Based on Recognizing Human Faces. PLoS ONE. 2014;9:e91708. https://doi:10.1371/journal.pone.0091708. [DOI] [PMC free article] [PubMed] [Google Scholar]