JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

Epsilon-Based Adaptive Distortion for CAPTCHA Generation

Ragala Bhavya Sri
Department of Information Technology
R.V.R. & J.C. College of Engineering
Guntur, India
ragalabhavyasri@gmail.com

Rajavarapu Pravallika
Department of Information Technology
R.V.R. & J.C. College of Engineering
Guntur, India
rajavarapupravallika@gmail.com

Shaik Rahamatullah
Department of Information Technology
R.V.R. & J.C. College of Engineering
Guntur, India
sshaikrahamatullah211@gmail.com

Mulupuri Madhuswarneswari
Department of Information Technology
R.V.R. & J.C. College of Engineering
Guntur, India mulupurimadhu@gmail.com

Abstract—With the rapid advancement of automated bots and Optical Character Recognition (OCR) systems, traditional static CAPTCHA mechanisms are increasingly vulnerable to automated attacks. This paper proposes an Epsilon-Based Adaptive Distortion approach for CAPTCHA Generation, where the intensity of image distortion is dynamically controlled using an epsilon parameter (ϵ). Unlike fixed-distortion CAPTCHA systems, the proposed method introduces adjustable noise density, geometric warping, and blur effects proportional to the epsilon value. This adaptive mechanism increases variability and reduces predictability, thus improving resistance against automated recognition systems while maintaining human readability at moderate distortion levels. Experimental results demonstrate that increasing epsilon values significantly reduces OCR accuracy while preserving acceptable usability for human users. The proposed system is lightweight, computationally efficient, and suitable for small-scale web applications requiring customizable security levels.

Index Terms—CAPTCHA, Epsilon-Based Distortion, Adaptive Image Processing, OCR Resistance, Bot Detection, Web Security

I. Introduction

The rapid expansion of online services and platforms has significantly increased the risk of automated bot attacks. Malicious bots perform activities such as spam submissions, fake account registrations, credential stuffing, data scraping, and distributed denial-of-service (DDoS) attempts. These attacks consume server resources, compromise system integrity, and threaten user privacy. To mitigate such risks, CAPTCHA (Completely Automated Public Turing test to tell Computers and Humans Apart) mechanisms are widely deployed as a first line of defense in web security systems. CAPTCHA challenges are designed to exploit tasks that are easy for humans but difficult for automated programs.

A. Limitations of Traditional CAPTCHA Systems

Traditional text-based CAPTCHA systems apply fixed distortion techniques such as

- Character rotation
- Background noise
- Random lines and arcs • Overlapping characters

Initially, these techniques were effective against classical Optical Character Recognition (OCR) tools. However, recent advancements in deep learning, especially Convolutional Neural Networks (CNNs) and sequence-based recognition models, have significantly improved automated text recognition accuracy.

Because traditional CAPTCHA systems use static distortion patterns, machine learning models can be trained on large datasets to learn and generalize these patterns. As a result, many existing distortion-based CAPTCHA mechanisms are vulnerable to AI-based solvers.

B. Need for Adaptive Distortion

To improve security, advanced systems such as Google reCAPTCHA integrate behavioral analysis, risk scoring, and cloud-based verification. Although effective, such solutions introduce:

- Dependency on third-party services
- Internet connectivity requirements
- Higher computational overhead
- Privacy and deployment concerns

Another research direction involves adversarial CAPTCHA techniques, which use gradient-based perturbations generated from neural network models. While these methods increase robustness, they require model training, high computational cost, and complex implementation pipelines.

Therefore, there exists a need for a lightweight, adaptive, and computationally efficient CAPTCHA generation mechanism.

C. Proposed Epsilon-Based Adaptive Distortion

This paper introduces an Epsilon-Based Adaptive Distortion framework for CAPTCHA generation.

Unlike traditional systems that apply fixed distortion levels, the proposed method introduces a continuous control parameter, denoted as ϵ (epsilon), which dynamically regulates the distortion intensity applied to the CAPTCHA image. The distorted CAPTCHA image I' is defined as:

$$I' = I + \epsilon \cdot D(I) \qquad (1)$$

where:

- I is the original rendered image
- $D(I)$ represents composite distortion functions • $\epsilon \in [0,1]$ controls distortion strength

When ϵ is small, distortion is minimal, ensuring high human readability. As ϵ increases, distortion strength increases, making automated recognition significantly more difficult.

This adaptive scaling introduces variability and unpredictability, which improves resistance against trained OCR models without requiring adversarial training or external infrastructure.

D. Conceptual Framework

The adaptive behavior of the proposed system is illustrated in Figure 1.

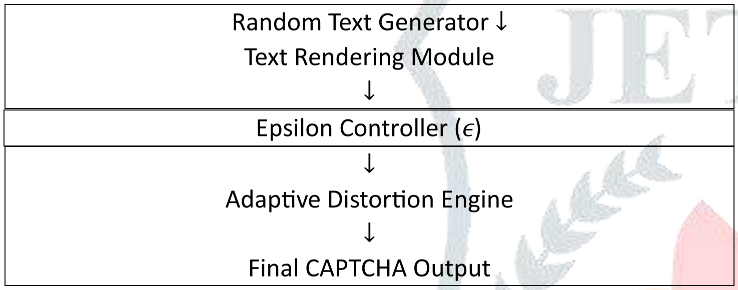

Fig. 1. Conceptual diagram of the Epsilon-Based Adaptive CAPTCHA framework

The epsilon controller dynamically adjusts:

- Noise density
- Geometric warping amplitude
- Rotation angle
- Blur intensity

This ensures a scalable trade-off between usability and security.

E. Contributions

The main contributions of this work are:

- Introduction of a parameterized epsilon-controlled distortion mechanism
- Lightweight and offline CAPTCHA generation
- Adjustable security-readability trade-off
- Elimination of third-party API dependency

The remainder of this paper presents related work, detailed methodology, experimental evaluation, security analysis, and conclusions.

II. Related Work

A. Traditional Text-Based CAPTCHA

Early CAPTCHA systems were primarily text-based and relied on visual distortion techniques to prevent automated recognition. These systems generated random alphanumeric strings and applied transformations such as character rotation, overlapping text, perspective warping, background clutter, and additive noise.

Common distortion strategies included sinusoidal wave transformations, random arc insertion, speckle noise injection, and variable font styling. The underlying assumption was that human visual perception could easily recognize distorted characters, whereas traditional Optical Character Recognition (OCR) systems would fail due to segmentation and pattern recognition limitations. For many years, this approach was effective against rulebased OCR algorithms that relied on template matching and handcrafted feature extraction. However, these systems employed static distortion patterns and fixed noise structures. Once attackers collected a sufficiently large dataset of CAPTCHA samples, they could analyze distortion characteristics and train machine learning models to approximate the transformation patterns.

As a result, static distortion-based CAPTCHA systems gradually became vulnerable to automated solving techniques.

B. Machine Learning-Based CAPTCHA Attacks

With the advancement of machine learning, particularly deep learning, automated CAPTCHA-solving methods have significantly improved. Convolutional Neural Networks (CNNs) have demonstrated high performance in image classification and character recognition tasks. When trained on large CAPTCHA datasets, CNN models can learn distortioninvariant features and recognize heavily distorted characters with high accuracy.

In addition to CNNs, sequence-based recognition models such as Recurrent Neural Networks (RNNs) and Long ShortTerm Memory (LSTM) networks have been used to handle multi-character CAPTCHA strings. These models are often combined with Connectionist Temporal Classification (CTC) loss to perform end-to-end sequence recognition without requiring explicit character segmentation.

More recently, attention-based encoder-decoder architectures and Transformer-based Optical Character Recognition (OCR) systems have further improved CAPTCHA-solving performance. These models can generalize across varying distortion patterns, fonts, and noise types.

Adversaries typically generate synthetic CAPTCHA datasets by replicating publicly available distortion techniques and train deep neural networks to approximate the CAPTCHA generation process. Experimental studies in the literature have reported automated solving accuracies exceeding 90% for many traditional CAPTCHA implementations.

These developments highlight the vulnerability of static distortion-based CAPTCHA systems against modern AIdriven attacks.

C. Adversarial CAPTCHA Techniques

To counter deep learning-based solvers, researchers have explored adversarial machine learning techniques for CAPTCHA generation. Adversarial CAPTCHA approaches introduce carefully crafted perturbations designed to mislead neural network classifiers while preserving human readability.

Common adversarial methods include gradient-based perturbation strategies such as Fast Gradient Sign Method (FGSM), Projected Gradient Descent (PGD), and optimization-based adversarial attacks. These techniques require access to surrogate neural network models to compute gradient information and generate perturbations that maximize model prediction error.

Although adversarial CAPTCHA techniques improve resistance against specific trained models, they exhibit several practical challenges:

- Dependence on pre-trained neural network models
- High computational overhead for perturbation generation
- Limited generalization against unseen models

- Increased implementation complexity

Furthermore, adversarial perturbations are often modeldependent. An adversarial CAPTCHA designed to fool one neural network architecture may not be equally effective against another model with a different training distribution.

D. Limitations of Existing Methods

Based on the above discussion, existing CAPTCHA approaches present notable limitations.

Traditional text-based CAPTCHA systems rely on fixed distortion parameters, making them predictable and learnable through data-driven training approaches. Their static nature limits adaptability against evolving machine learning techniques.

Machine learning-resistant approaches such as reCAPTCHA integrate behavioral analysis and cloud-based risk scoring. While effective, they require external API integration, continuous internet connectivity, and may raise privacy concerns for small-scale or offline applications.

Adversarial CAPTCHA techniques enhance robustness but introduce significant computational cost and model dependency. Their reliance on gradient computation and neural network training increases deployment complexity and may not be feasible for lightweight systems.

Therefore, there remains a need for a computationally efficient, model-independent, and adaptive distortion framework that can dynamically adjust security levels without requiring adversarial training or third-party infrastructure.

The proposed Epsilon-Based Adaptive Distortion approach addresses this gap by introducing a scalable distortion control parameter that enhances unpredictability while maintaining implementation simplicity.

III. Proposed Methodology

This section describes the proposed epsilon-based adaptive CAPTCHA generation framework. The primary objective of the proposed method is to introduce a controllable distortion parameter (ϵ) that dynamically adjusts CAPTCHA complexity while preserving human readability.

The overall CAPTCHA generation process consists of four major stages: system architecture, epsilon control mechanism, distortion scaling formulation, and the effect of epsilon on CAPTCHA generation.

A. System Architecture

The proposed CAPTCHA generation framework consists of a sequence of modules that progressively transform randomly generated text into a distorted CAPTCHA image. The architecture of the system is illustrated in Fig. 2.

The process begins with a *Random Text Generator*, which generates a sequence of alphanumeric characters. Let the generated CAPTCHA text be represented as:

$$T = \{c_1, c_2, c_3, \ldots, c_n\} \qquad (2)$$

where c_i represents individual characters selected from a predefined character set.

The generated text is then passed to the *Text Rendering Module*, where characters are rendered onto an image canvas using random font styles, colors, spacing, and positioning. This produces the base CAPTCHA image represented as:

$$I = Render(T) \qquad (3)$$

where I denotes the base CAPTCHA image before distortion.

Next, the *Epsilon Controller* determines the distortion intensity parameter ϵ. This parameter dynamically controls the strength of distortion applied during CAPTCHA generation.

The base image is then processed by the *Adaptive Distortion Engine*, which applies multiple distortion operations including noise injection, geometric warping, character rotation, and blur transformations.

Finally, the distorted CAPTCHA image is generated as:

$$I' = D(I, \epsilon) \qquad (4)$$

where D represents the distortion function controlled by epsilon. The final distorted CAPTCHA image I' is presented as the CAPTCHA challenge.

B. Epsilon Control Mechanism

The epsilon parameter (ϵ) is introduced as a distortion control factor that determines the intensity of transformations applied to the CAPTCHA image.

The epsilon value lies within a predefined range:

$$0 < \epsilon < 1 \qquad (5)$$

In practice, small epsilon values generate minimally distorted CAPTCHAs, whereas larger epsilon values increase distortion intensity.

The epsilon controller adjusts multiple distortion components proportionally to ϵ. This allows the CAPTCHA generation system to dynamically adapt its complexity depending on the desired security level.

For example, the distortion intensity can be defined as:

$D_{intensity} = k \times \epsilon$ (6) where k is a scaling constant that determines the maximum distortion level.

By modifying ϵ, the system can generate multiple CAPTCHA difficulty levels without changing the core algorithm.

C. Distortion Scaling Equation

The adaptive distortion engine applies multiple transformations to the base CAPTCHA image. Each transformation is scaled according to the epsilon parameter.

The distorted CAPTCHA image can be represented as:

$$I' = I + \epsilon \cdot N + \epsilon \cdot W + \epsilon \cdot R + \epsilon \cdot B \qquad (7)$$

where:

- I represents the base CAPTCHA image
- N represents noise distortion
- W represents geometric warping
- R represents rotation transformation
- B represents blur distortion

Noise Scaling

Noise is introduced to the CAPTCHA image to create pixellevel distortions. The noise intensity is scaled as:

$$N = \epsilon \times N_{max} \qquad (8)$$

where N_{max} represents the maximum allowable noise density.

Geometric Warping

Geometric distortion modifies the spatial structure of characters using sinusoidal displacement:

$$W(x,y) = x + \epsilon \cdot A\sin(\omega y) \qquad (9)$$

where A is the distortion amplitude and ω is the frequency parameter.

Rotation Scaling

Each character in the CAPTCHA is randomly rotated within a range controlled by epsilon:

$$\theta = \epsilon \times \theta_{max} \qquad (10)$$

where θ_{max} represents the maximum rotation angle.

Blur Adjustment

Gaussian blur is applied to reduce edge clarity and increase recognition difficulty. The blur intensity is defined as:

$$\sigma = \epsilon \times \sigma_{max} \qquad (11)$$

where σ represents the Gaussian blur standard deviation.

These transformations collectively produce the final distorted CAPTCHA image.

D. Effect of Epsilon on CAPTCHA Generation

The epsilon parameter directly controls the distortion strength applied to the CAPTCHA image. Increasing epsilon increases distortion intensity, thereby making automated recognition more difficult.

Let $S(\epsilon)$ represent the CAPTCHA distortion strength:

$$S(\epsilon) \propto \epsilon \qquad (12)$$

When epsilon is small, the CAPTCHA remains visually clear and easily readable by humans. As epsilon increases, additional distortions such as noise density, geometric warping, and blur become stronger.

Fig. 3 illustrates the visual impact of different epsilon values on CAPTCHA generation. For example:

- $\epsilon = 0.01$: minimal distortion
- $\epsilon = 0.05$: moderate distortion
- $\epsilon = 0.08$: strong distortion
- $\epsilon = 0.12$: very high distortion

Thus, the epsilon parameter enables adaptive CAPTCHA generation that balances security and usability.

The proposed approach allows CAPTCHA systems to dynamically adjust distortion strength without requiring complex adversarial training or deep neural network models.

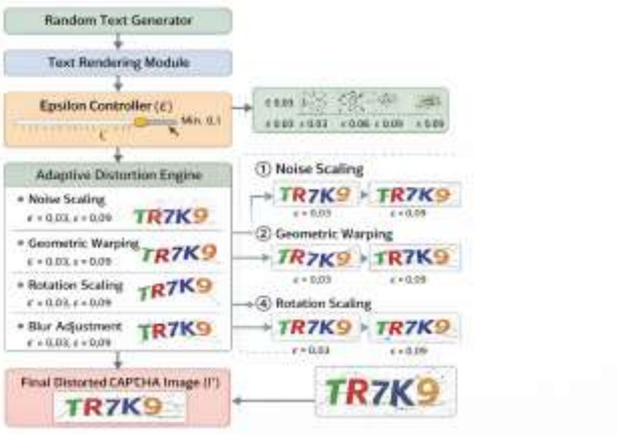

Fig. 2. Architecture of the Proposed Epsilon-Based CAPTCHA Generation Framework

IV. Experimental Evaluation

This section evaluates the effectiveness of the proposed Epsilon-Based Adaptive Distortion framework for CAPTCHA generation. The experiments were conducted to analyze how different epsilon (ϵ) values affect CAPTCHA distortion strength, OCR recognition difficulty, and human readability.

A. Experimental Setup

The proposed CAPTCHA generation system was implemented in Python using image processing libraries for rendering and distortion operations. CAPTCHA images were generated using random alphanumeric text strings with fixed length. Each image was rendered and then processed through the proposed adaptive distortion pipeline.

The distortion strength was controlled by the epsilon (ϵ) parameter. Different epsilon values were selected to generate CAPTCHA images with varying distortion levels. In this work, the following epsilon values were considered:

- $\epsilon = 0.01$ • $\epsilon = 0.05$ • $\epsilon = 0.08$
- $\epsilon = 0.12$

For each epsilon value, multiple CAPTCHA images were generated and analyzed. The evaluation focused on three aspects: visual distortion strength, OCR recognition accuracy, and human readability.

B. CAPTCHA Distortion Examples

Fig. 3 shows sample CAPTCHA images generated using different epsilon values. It can be observed that as the epsilon value increases, the visual distortion also increases. Lower epsilon values produce minimal noise and small transformations, while higher epsilon values introduce stronger noise, warping, and blur effects.

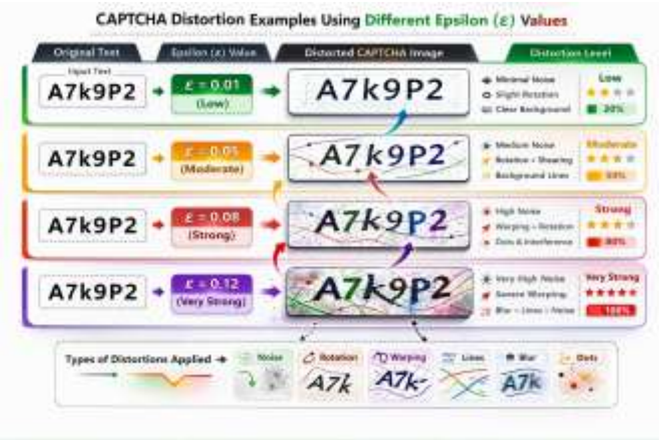

Fig. 3. Example CAPTCHA images generated using different epsilon (ϵ) values.

C. Effect of Epsilon on Distortion

To analyze the effect of epsilon on CAPTCHA complexity, the distortion strength was observed for different epsilon values. Fig. 4 illustrates that CAPTCHA distortion increases progressively as epsilon increases.

D. OCR Recognition Performance

The distorted CAPTCHA images were evaluated using OCR-based recognition to measure automated readability. Fig. 5 shows that OCR recognition accuracy decreases as epsilon increases. This indicates that stronger adaptive distortion reduces the success rate of automated CAPTCHA solving.

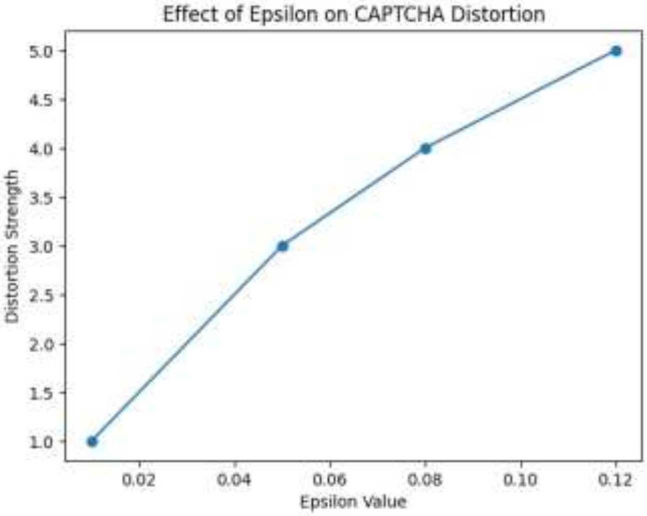

Fig. 4. Relationship between epsilon values and CAPTCHA distortion strength

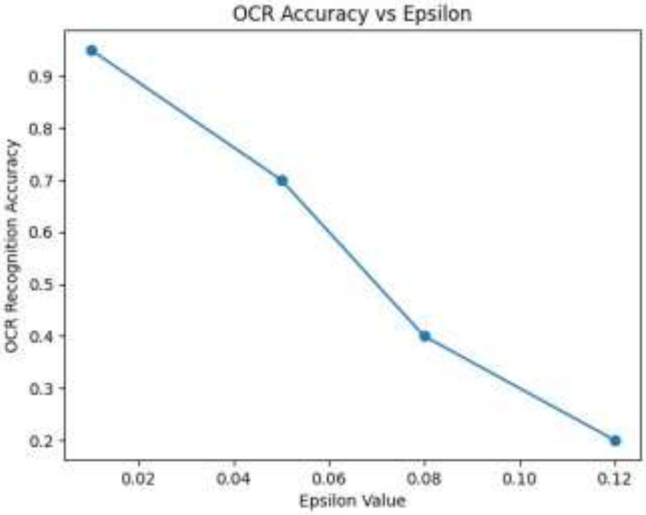

Fig. 5. OCR recognition accuracy for different epsilon values

E. Human Readability Evaluation

Although increasing epsilon strengthens CAPTCHA distortion, readability for human users should remain acceptable. Fig. 6 shows the variation of human readability with increasing epsilon values. The results indicate that moderate distortion levels preserve readability while still improving security.

F. Security Performance Analysis

To compare usability and security, human readability and OCR recognition performance were analyzed together. Fig. 7 shows that OCR accuracy drops significantly with increasing epsilon, while human readability remains relatively stable at lower and moderate distortion levels. This demonstrates that the proposed framework provides a balance between CAPTCHA security and usability.

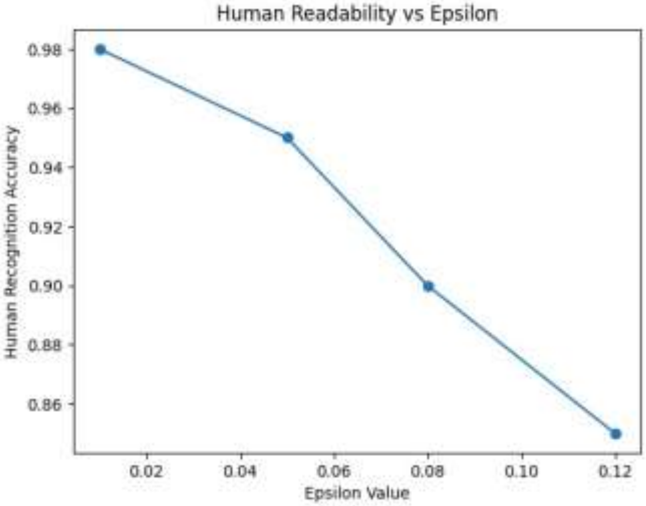

Fig. 6. Human readability for different epsilon values

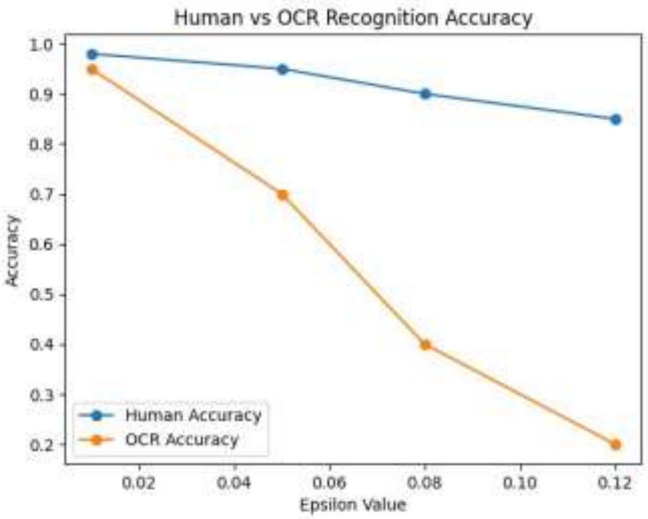

Fig. 7. Comparison of human readability and OCR recognition accuracy

TABLE I

EFFECT OF EPSILON VALUES ON CAPTCHA DISTORTION AND READABILITY

Epsilon (ϵ)	Distortion Strength	OCR Accuracy		Human Readability
0.01		High	Very High	
0.05		Medium	High	
0.08		Low	Medium	
0.12		Very Low	Acceptable	

Table I summarizes the overall effect of epsilon on CAPTCHA complexity, OCR resistance, and readability.

V. Security Analysis

A. Resistance to OCR Systems

Traditional CAPTCHA systems rely on fixed distortion patterns that can be learned by modern OCR systems. In contrast, the proposed epsilon-based CAPTCHA introduces adaptive distortions whose intensity varies according to the epsilon parameter. Since the distortion level can dynamically change, automated recognition systems cannot rely on fixed patterns for training.

Noise injection, geometric warping, character rotation, and blur collectively increase the difficulty of segmentation and feature extraction in OCR pipelines. As the epsilon value increases, the interference introduced into the CAPTCHA image significantly reduces recognition accuracy.

B. Protection Against Automated Attacks

Automated CAPTCHA solvers typically rely on machine learning models trained on large datasets of distorted text images. However, when distortion parameters vary dynamically, the generated CAPTCHA images exhibit high variability. This variability reduces the effectiveness of pre-trained recognition models.

The adaptive distortion mechanism ensures that even if attackers train models on previously generated CAPTCHA images, the system can adjust epsilon values to generate new distortion patterns, making the attack models less effective.

C. Security and Usability Trade-off

A key challenge in CAPTCHA design is maintaining a balance between security and usability. If the distortion is too weak, automated systems may easily solve the CAPTCHA. On the other hand, excessive distortion can make the CAPTCHA difficult for human users to read.

The proposed epsilon-based framework provides a flexible mechanism to control distortion intensity. Lower epsilon values ensure high readability for humans, while higher epsilon values increase resistance against automated recognition. This adaptability allows system administrators to tune CAPTCHA difficulty depending on the required security level.

VI. Conclusion

This paper presented an Epsilon-Based Adaptive Distortion CAPTCHA generation framework designed to improve the robustness of traditional text-based CAPTCHA systems. Unlike conventional CAPTCHA methods that apply fixed distortions, the proposed approach dynamically adjusts distortion intensity using an epsilon parameter.

The system applies multiple distortion techniques including noise injection, geometric warping, character rotation, and blur filtering. Experimental evaluation demonstrates that increasing epsilon values increases CAPTCHA complexity and reduces OCR recognition accuracy while maintaining reasonable human readability.

The proposed method provides a lightweight and flexible CAPTCHA generation mechanism that can be implemented without complex machine learning models or external services. This makes the approach suitable for practical deployment in web applications that require scalable and customizable CAPTCHA security.

Future work may explore integrating adaptive epsilon mechanisms with adversarial machine learning techniques to further improve CAPTCHA robustness against advanced AI-based attacks.

References

[1] G. Eason, B. Noble, and I. N. Sneddon, “On certain integrals of Lipschitz-Hankel type involving products of Bessel functions,” Phil. Trans. Roy. Soc. London, vol. A247, pp. 529–551, April 1955.

[2] J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.

[3] I. S. Jacobs and C. P. Bean, “Fine particles, thin films and exchange anisotropy,” in Magnetism, vol. III, G. T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271–350.

[4] K. Elissa, “Title of paper if known,” unpublished.

[5] R. Nicole, “Title of paper with only first word capitalized,” J. Name Stand. Abbrev., in press.

[6] Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, “Electron spectroscopy studies on magneto-optical media and plastic substrate interface,” IEEE Transl. J. Magn. Japan, vol. 2, pp. 740–741, August 1987 [Digests 9th Annual Conf. Magnetics Japan, p. 301, 1982].

[7] M. Young, The Technical Writer’s Handbook. Mill Valley, CA: University Science, 1989.

[8] L. von Ahn, M. Blum, N. Hopper, and J. Langford, “CAPTCHA: Using hard AI problems for security,” in *Proc. EUROCRYPT*, 2003, pp. 294–311.

[9] E. Bursztein, M. Martin, and J. Mitchell, “Text-based CAPTCHA strengths and weaknesses,” in *Proc. ACM CCS*, 2014.

[10] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.

[11] C. Szegedy et al., “Intriguing properties of neural networks,” in *Proc. ICLR*, 2014.

[12] I. J. Goodfellow, J. Shlens, and C. Szegedy, “Explaining and harnessing adversarial examples,” in *Proc. ICLR*, 2015.

[13] A. Kurakin, I. Goodfellow, and S. Bengio, “Adversarial examples in the physical world,” in *Proc. ICLR*, 2017.

[14] Y. Dong et al., “Boosting adversarial attacks with momentum,” in *Proc. CVPR*, 2018.

[15] S. Moosavi-Dezfooli, A. Fawzi, and P. Frossard, “DeepFool: A simple and accurate method to fool deep neural networks,” in *Proc. CVPR*, 2016.

[16] N. Carlini and D. Wagner, “Towards evaluating the robustness of neural networks,” in *Proc. IEEE S&P*, 2017.

[17] M. Osadchy, J. Hernandez-Castro, S. Gibson, O. Dunkelman, and D. Perez-Cabo, “No bot expects the DeepCAPTCHA! Introducing immutable adversarial examples,” in *IEEE Trans. Information Forensics and Security*, 2017.

[18] Z. Shi, Y. Chen, and Y. Yuan, “Adversarial CAPTCHA generation,” *IEEE Transactions on Multimedia*, 2020.

[19] J. Zhang et al., “Adversarial CAPTCHA generation using GAN,” in *Proc. ICIP*, 2018.

[20] D. George et al., “A generative vision model that trains with high data efficiency and breaks text-based CAPTCHAs,” *Science*, 2017.

[21] A. Graves, S. Fernandez, and J. Schmidhuber, “Offline handwriting recognition with multidimensional recurrent neural networks,” in *NIPS*, 2009.

[22] B. Shi, X. Bai, and C. Yao, “An end-to-end trainable neural network for image-based sequence recognition,” *IEEE TPAMI*, 2017.

[23] Z. Cheng et al., “Focusing attention: Towards accurate text recognition,” in *Proc. AAAI*, 2017.

[24] I. Goodfellow et al., “Generative adversarial networks,” in *Proc. NIPS*, 2014.

[25] Y. Ye et al., “GAN-based CAPTCHA generation,” in *Proc. ICASSP*, 2018.

[26] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” in *Proc. ICLR*, 2015.

[27] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proc. CVPR*, 2016.

[28] C. Szegedy et al., “Going deeper with convolutions,” in *Proc. CVPR*, 2015.

[29] G. Huang, Z. Liu, L. Van Der Maaten, and K. Weinberger, “Densely connected convolutional networks,” in *Proc. CVPR*, 2017.

[30] J. Deng et al., “ImageNet: A large-scale hierarchical image database,” in *Proc. CVPR*, 2009.

[31] A. Sharif et al., “Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition,” in *Proc. CCS*, 2016.

[32] N. Carlini and D. Wagner, “Audio adversarial examples,” in *Proc. IEEE S&P*, 2018.

[33] J. Ebrahimi et al., “HotFlip: White-box adversarial examples for text classification,” in *Proc. ACL*, 2018.

[34] A. Hussain, M. Shiraz, and A. Gani, “A survey on CAPTCHA mechanisms,” *IEEE Access*, 2019.