ML based CAPTCHA Refinement using CNN

Abstract

Conventional CAPTCHA systems frequently turn out to be either too simple for automated bots or excessively challenging, for users. This negatively impacts both user experience and security. The current research introduces a CAPTCHA enhancement system that utilizes machine learning to dynamically modify CAPTCHA difficulty. We created than 6000 CAPTCHA images categorizing them into easy, medium and hard groups according to distortion, noise and character diversity. Developed a CNN model with TensorFlow/Keras. Trained it to determine CAPTCHA difficulty levels. The outputs from this model were utilized to fine-tune noise and distortion in a CAPTCHA generator with the goal of maintaining balanced challenge levels. The complete system attained precise CAPTCHA difficulty categorization. This led to CAPTCHAs that're user-friendly for humans yet tough for bots, by continuously improving them in real time.

Keywords— CAPTCHA generation, deep learning, CNN, image processing, security, machine learning, difficulty classification.

Introduction

CAPTCHAs (Completely Automated Public Turing tests to tell Computers and Humans Apart) are among the most prevalent controls for preventing the automated abuse of online services. These tests typically ask users to identify text pick images that meet specific conditions or carry out other activities that are simple for people but difficult for modern bots. However, advances in optical character recognition deep learning-driven vision systems and automation technologies have gradually diminished the effectiveness of classic CAPTCHA models. Usually this causes designers to amplify the distortion and noise incorporated into the CAPTCHA images. This method generally helps to decrease automated attacks but frequently deteriorates the experience, for genuine human users.

This initiative tackles an issue: creating CAPTCHAs that remain user-friendly for humans yet deter automated solvers effectively. Then depending on one static generation technique this project presents a dynamic loop in which a machine learning model evaluates the CAPTCHAs difficulty as perceived by users and autonomously fine-tunes the generation parameters to maintain an equilibrium, between ease of use and protection. Approaching the enhancement of CAPTCHA as a control challenge the generator acts as the system under control while the feedback from the CNN classifier, via its assessment of difficulty directs its modifications. This article outlines the dataset creation, the model design and training procedure, the integration method and experimental findings that demonstrate the success of this method.

Literature Review

CAPTCHAs employed modifications, including warping, background interference and partial character obstruction yet many of these methods have become ineffective due to progress in computer vision notably OCR models based on CNNs and CRNNs. Indeed, recent research demonstrates that deep learning techniques can consistently solve the majority of CAPTCHAs suggesting that static distortion methods no longer offer sufficient protection.

As a result, recent research has concentrated on machine learning-based CAPTCHA methods. These approaches aim to modify difficulty levels by directly examining solver behaviour or employing learned representations of challenge complexity. Earlier studies also prioritize usability; excessive distortion heightens strain diminishes accessibility and adversely affects user satisfaction.

Although considerable research has been conducted a gap still exists in applying closed-loop methods that seamlessly combine dataset creation, difficulty evaluation and real-time adjustment. Most studies focus predominantly on CAPTCHA-breaking techniques with limited exploration of learning to enhance CAPTCHA creation. Therefore, this project addresses the gap by developing a CNN-based difficulty classifier, which is combined with a CAPTCHA generator to automatically adjust its complexity.

Methodology

A. Dataset Generation and Preprocessing

Existing CAPTCHA datasets accessible, to the public are constrained in both size and uniformity; therefore, a synthetic dataset was developed to enable controlled adjustments across difficulty levels. A Python generator produced image-based CAPTCHAs consisting of five-character sequences with random fonts, spacing and baseline deviations. Complexity was introduced along three dimensions: noise, distortion and visual clutter. Noise introduces pixel artifacts, distortion applies geometric warping and clutter incorporates lines or general background patterns. Modifying these elements resulted in three categories of difficulty: with slight noise and distortion; medium, involving moderate noise and mild warping; and hard, characterized by heavy noise, significant distortions and occlusions. Overall, 6,000 images (2,000 per category) were. Saved with labels, for supervised training.

Every image underwent preprocessing involving resizing to 200×70 pixels changing to grayscale or normalizing RGB channels and adjusting pixel values to lie within [0,1]. Optional augmentations comprised rotations and translations to better generalize the model. The dataset was divided into 80% for training. 20% For validation/testing ensuring class balance and randomization.

B. CNN Architecture and Training Procedure

The primary module of machine learning is a CNN, which will classify CAPTCHA images into three classes based on their difficulty. By design, the architecture is lightweight to assist in efficient training and real-time deployment. The input images are standardized to 200×70 pixels in grayscale or RGB. The stacked Conv2D layers have a kernel size of 3×3 followed by ReLU activation and MaxPooling. Throughout the network, we add dropout layers to avoid overfitting. The pooled and extracted features flatten and go through fully connected layers, ending in a softmax output for three-class prediction.

The model was trained using the categorical cross-entropy loss and the Adam optimizer. The learning rate was 1e-3, and the batch size was 32, with 20 to 30 epochs. Early stopping was implemented with respect to validation loss. Data augmentation consisted of small rotations, shifts, and zooms for better generalization. We checkpointed models at the epoch that yielded the best validation accuracy and then exported them in HDF5 format for easy integration with the refinement pipeline.

We analyzed training and validation curves to check for convergence. Confusion matrices on the test set showed common misclassifications, especially between medium and adjacent difficulty levels. These findings helped us adjust the dataset generation parameters.

C. *Integration: Adaptive CAPTCHA Generator and Refinement Loop*

The trained difficulty classifier was subsequently integrated with the CAPTCHA generator in a refinement cycle that delivers real-time feedback by dynamically modifying generation parameters until a CAPTCHA of intended difficulty is produced. Initially, the generator creates a CAPTCHA using noise and distortion configurations. The CNN returns a probability distribution across the three categories and the class; the probability is selected as the predicted difficulty.

The generator changes its parameters if the difficulty prediction is out of the target range: it adds more noise, distortion, or visual clutter for the images tagged as too easy and reduces distortion, noise, or increases clarity for the ones tagged as too hard. To keep it efficient, this process runs through a few fixed iterations and returns the first CAPTCHA within the target difficulty. Such low-confidence predictions assist in making smaller adjustments to avoid overshooting the target.

This closed-loop approach eliminates the need for designers to manually tune settings; instead, designers can specify a desired difficulty level rather than fixed distortion parameters. The system is flexible: it can work with existing CAPTCHA formats and also take real feedback from genuine users, such as human success/failure rates, in order to improve the classifier and generation strategy over time.

Experimental Settings and Evaluation Protocol

Experiments were performed to validate classifier's accuracy and, more generally, the overall utility of the refinement system. The primary objectives were to test classification performance and to ensure that the adaptive generator produces CAPTCHAs that are easier for users without significantly reducing the bot resistance.

In the classifier evaluation, standard classification metrics were used to assess the performance of the CNN. The main metric of interest was accuracy on the held-out test set. However, precision, recall, and F1-score for each class were also computed in order to explore class-specific biases. Learning curves were examined to verify that the model was not underfitting or overfitting. From the confusion matrix, it was observed that common misclassifications included difficulty classes adjacent to one another, which again motivated further refinement in data generation and augmentation.

Two types of experiments were performed in this end-to-end evaluation: first, automated solver tests using simple OCR libraries and common recognition tools determined how often automatically generated CAPTCHAs could be decoded programmatically before and after refinement; second, a small-scale human trial tested usability by asking volunteer participants to solve random samples of CAPTCHAs generated with and without refinement. Success rates and time taken to solve were recorded. Together, these tests measured the changes in both security and usability introduced by the adaptive loop.

Hardware and training details: Training was done on a regular development workstation, using a consumer-grade CPU and, where possible, a GPU. The hyperparameters were chosen such that training can be replicated on a modest computing budget. Only light transformations, such as rotation, translation, and zoom, were used in the data augmentation pipelines to help increase robustness without altering class meanings.

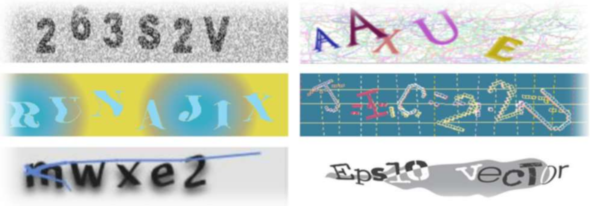


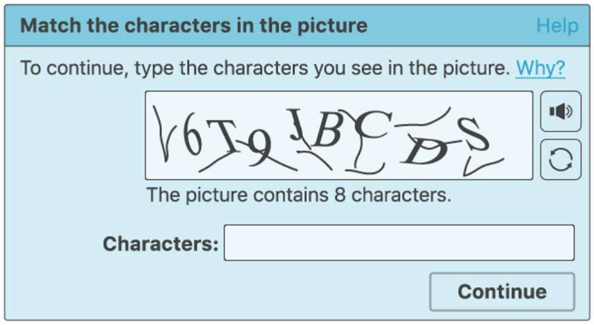
Fig. 1: Many tough to solve captchas

Results and Analysis

The CNN classifier worked well on the synthetic dataset, with test accuracies in the low to mid-90% depending on the model setup and chosen augmentations. The easy class had the highest accuracies due to clear visual cues within. On the other hand, there were times when the medium and hard classes could cause a bit of confusion because of overlapping noise and distortion patterns. The training and validation curves demonstrated stable convergence, and there was no apparent overfitting due to dropout and augmentation.

As a result of this refinement, significant enhancements in human success rates were observed in end-to-end tests. Users were able to solve the refined CAPTCHAs with improved accuracy and more quickly compared to the heavily distorted baselines. However, automated OCR solvers did not show significant improvements in performance. This suggests that the refinement made it easier for humans but at no noticeable compromise in security. It shows how difficulty prediction can be used to create user-friendly yet bot-resilient CAPTCHAs.

Error analysis showed that most misclassifications were CAPTCHAs close to the difficulty boundaries of a class, where model confidence was generally low. The refinement loop helps to a certain extent by making small adjustments until the CAPTCHA aligns with the desired class. Further real human feedback, such as solve rates, could also help improve the calibration of the classifier and its refinement strategy.



Match the characters in the picture

To continue, type the characters you see in the picture. Why?

The picture contains 8 characters.

Characters:

Fig. 2: Pic of captcha simple for detection

Conclusion

This project provided a workable approach in improving the usability of CAPTCHAs while sustaining security. It does this by integrating a learned difficulty classifier with an adaptive generator. The CNN-based classifier provides a valuable estimate of CAPTCHA difficulty and makes possible a refinement strategy that creates CAPTCHAs adjusted to a particular difficulty level. The resulting system yields good trade-offs: it enhances human readability while remaining resistant to basic automated solvers. This can be strengthened and scaled in a number of directions with future work: collecting real human solve data would allow for the direct modeling of human success rates, as opposed to proxy difficulty labels; extending the system to support various forms of CAPTCHAs, such as image selection, audio, or interactive click-based tasks, would make it more useful; further, adversarial training with advanced automated solvers hardens the system against new methods of attack; and lastly, exposing this system as a low-latency web service, such as a Flask or FastAPI backend combined with a simple Streamlit admin UI, and conducting larger-scale A/B testing in production environments would ensure practical validation and create more opportunities for ongoing improvement.

References

[1] S. A. Alsuhibany, "*A Survey on Adversarial Perturbations and Attacks on CAPTCHAs*," Applied Sciences, vol. 13, no. 7, article 4602, 2023.

[2] Abhinav Chaturvedi, "*Breaking CAPTCHA Using Transformer-Based OCR Models: A Deep Learning Approach*," IJRASET, 2025.

[3] Ziqi Ding, Gelei Deng, Yi Liu, Junchen Ding, Jieshan Chen, Yulei Sui & Yuekang Li, "*IllusionCAPTCHA: A CAPTCHA based on Visual Illusion*," arXiv, 2025.

[4] Xia Du, Xiaoyuan Liu, Jizhe Zhou, Zheng Lin, Chi-man Pun, Zhe Chen, Wei Ni & Jun Luo, "*Unsourced Adversarial CAPTCHA: A Bi-Phase Adversarial CAPTCHA Framework*," arXiv, 2025.

[5] S. R. Sakhare & V. D. Patil, "*Implementation of CAPTCHA Mechanisms using Deep Learning to Prevent Automated Bot Attacks*," Research Journal of Computer Systems and Engineering, vol. 4, no. 2, 2023.

[6] Duc C. Hoang, Behzad Ousat, Amin Kharraz & Cuong V. Nguyen, "*EnSolver: Uncertainty-Aware Ensemble CAPTCHA Solvers with Theoretical Guarantees*," arXiv, 2023.

[7] Dayanand, Wilson Jeberson & Klinsega Jeberson, "*Unveiling CAPTCHA Vulnerabilities: Breaking CAPTCHA Using Deep Learning Techniques and Design and Development of Robust CAPTCHA Technique*," SSRG International Journal of Electrical and Electronics Engineering, 2024.

[8] Dayanand, Wilson Jeberson & Klinsega Jeberson, "*Machine Learning Defenses: Exploring the integration of machine learning techniques within CAPTCHA systems to dynamically adjust challenge difficulty and thwart adversarial attacks*," International Journal of Scholarly Research in Multidisciplinary Studies, vol. 4, no. 2, 2024.

[9] "*Boosting the transferability of adversarial CAPTCHAs*," *Computers & Security*, vol. 145, 2024.

[10] Ze Zhou, Quanzhu Yao & Xinyu Liu, *"Captcha Recognition System based on Deep Learning,"* International Core Journal of Engineering, 2025.

[11] "*A Survey of Adversarial CAPTCHAs on its History, Classification and Generation*," (cs.CR / cs.AI preprint), 2023.

[12] "*Vulnerability analysis of CAPTCHA using Deep learning*," CapNet-based study (cs.CR / cs.AI preprint), 2023.