iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 13 Issue: XI Month of publication: November 2025

DOI: https://doi.org/10.22214/ijraset.2025.75241

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue XI Nov 2025- Available at www.ijraset.com

Breaking CAPTCHA Using Transformer-Based OCR Models: A Deep Learning Approach

Abhinav Chaturvedi
India

Abstract: CAPTCHA (Completely Automated Public Turing test to tell Computers and Humans Apart) systems are widely employed to secure web applications against automated bots. However, recent advancement in deep learning and pattern recognition, particularly transformer architectures, have posed significant challenges to the robustness of CAPTCHA security measures. The objective of this research is to analyze security flaws even in highly distorted and complex character CAPTCHAs. In this research, we propose a novel approach using Optical Character Recognition (OCR) models based on Convolutional Recurrent Neural Networks (CRNN) and hybrid CNN-Transformer architectures to evaluate their efficacy in breaking CAPTCHA images. In this paper, two distinct OCR models have been trained: a baseline CRNN model and an advanced hybrid CRNN-Transformer model. Extensive experiments demonstrated that the hybrid CNN-Transformer significantly outperformed the traditional CRNN model, achieving an accuracy of 95.1%, compared to the baseline CRNN model's 79.3%. Findings from the research highlight the susceptibility of current CAPTCHA systems to transformer-based OCR techniques and suggest that integrating transformer components into OCR models markedly enhances their capability in recognizing distorted text, posing critical implications for future CAPTCHA design and cybersecurity.

Keywords: CAPTCHA, Transformers, Optical Character Recognition, Deep Learning, Convolutional Recurrent Neural Networks.

I. INTRODUCTION

CAPTCHA (Completely Automated Public Turing Test to Tell Computers and Humans Apart) [1] is a widely employed security measure designed to protect online services like payment, web surfing from malicious automated bot activities. Historically, Text based CAPTCHAs have relied on visual complexities, including distorted characters, noisy backgrounds, and varying degrees of character deformation, to differentiate human users from automated systems. These challenges are carefully crafted to be solvable by humans but exceedingly difficult for automated algorithms. Nonetheless, recent advancements in artificial intelligence, particularly within the realm of deep learning], have significantly weakened the effectiveness of traditional CAPTCHA methodologies. Specifically, convolutional neural networks (CNNs) [2,3] and recurrent neural networks, such as Long Short-Term Memory (LSTM) [4-6] models, have substantially improved the accuracy and efficiency of automated CAPTCHA-solving systems. These models, which combine spatial feature extraction and sequential processing, have successfully managed to recognize and interpret highly distorted textual data traditionally considered secure from automation.

The recent introduction and widespread adoption of Transformer architectures [7,8] have further revolutionized this field. Transformer models utilize sophisticated self-attention mechanisms to effectively capture long-range dependencies in sequential data, outperforming previous recurrent architectures. When combined with CNN-based feature extraction, these Transformer-enhanced hybrid models present a formidable approach to OCR tasks, particularly in complex scenarios involving heavily distorted and noisy CAPTCHA images. In this study, we comprehensively explore the application and effectiveness of Transformer-based OCR models for breaking CAPTCHA security. Our research involves training and evaluating two distinct architectures: a conventional Convolutional Recurrent Neural Network (CRNN) model, serving as a baseline, and a hybrid CNN-Transformer model. The basic CRNN architecture incorporates convolutional layers for feature extraction followed by bidirectional LSTM layers, optimized using AdamW and Focal CTC Loss for improved convergence and robustness. The advanced Hybrid CNN-Transformer architecture integrates CNN-based feature extraction with Transformer encoder layers, employing AdamW optimization and standard CTC Loss to effectively capture sequential dependencies.

By systematically comparing these models, we demonstrate that the hybrid Transformer-integrated architecture substantially outperforms the traditional CRNN model in terms of accuracy and robustness. Furthermore, this investigation underscores critical security implications arising from the enhanced capabilities of modern OCR systems, emphasizing an urgent need for advanced CAPTCHA designs resilient to rapidly evolving automated threats.

II. RELATED WORK

Early CAPTCHA recognition techniques primarily relied on traditional Optical Character Recognition (OCR) methods and handcrafted feature extraction, such as character segmentation, statistical modeling, pixel density analysis, and pattern matching techniques [9, 11]. The foundational idea of CAPTCHA was proposed by von Ahn et al. [1], designed explicitly to differentiate human users from automated bots by presenting tasks that are straightforward for humans but computationally challenging for automated algorithms. Despite their initial effectiveness, these traditional methods struggled with variations in character distortions, noise, and other complexities intentionally introduced to secure CAPTCHA images [10].

The emergence of deep learning marked a significant turning point in CAPTCHA recognition research. Convolutional Neural Networks (CNNs), initially introduced by LeCun et al. [12] for handwritten digit recognition, were later extended to recognize complex scene texts and distorted CAPTCHA characters with remarkable effectiveness. Goodfellow et al. [2] demonstrated the efficacy of deep CNN architectures in handling multi-digit recognition from street-view imagery. Further advancement by Jaderberg et al. [3] introduced robust techniques that significantly enhanced the recognition of complex and varied textual images, inspiring subsequent research into CAPTCHA-solving methods.

Additionally, the integration of recurrent neural networks (RNNs), specifically Long Short-Term Memory (LSTM) units developed by Hochreiter and Schmidhuber [4], improved sequential data modeling capabilities. These models effectively captured contextual information within text sequences, greatly enhancing the accuracy of CAPTCHA recognition tasks. Subsequent research by Ranzato et al. [5], Lipton et al. [6], and Challagundla et al. [13] demonstrated substantial performance improvements by utilizing hybrid CNN-LSTM models.

More recently, Transformer architectures have significantly advanced the field of sequence modeling. Introduced by Vaswani et al. [7], Transformers employ self-attention mechanisms capable of modeling long-range dependencies without the recurrent structure limitations. Initially popularized in natural language processing tasks, Transformer-based models have been adapted for various computer vision applications by Dosovitskiy et al. [8], who demonstrated their powerful image-recognition capabilities. Subsequent studies have extensively evaluated the potential of Transformers in OCR tasks, further supported by comprehensive surveys reviewing their advancements and potential in visual recognition [14].

Recent research has emphasized hybrid models combining CNN-based feature extraction with Transformer encoders, showcasing superior performance in CAPTCHA recognition scenarios compared to standalone CNN and CNN-LSTM architectures [10, 16, 17]. Specific studies by Wan et al. [15] and Challagundla et al. [13] demonstrated the effectiveness of hybrid architectures in achieving higher recognition accuracy on various CAPTCHA datasets.

Multiple comprehensive reviews and surveys have systematically assessed developments in CAPTCHA-breaking techniques, underscoring their strengths, vulnerabilities, and evolving security challenges posed by advanced deep learning methods [18, 19]. Concurrent research on adversarial attacks and security evaluations [20- 23] further highlights critical vulnerabilities in CAPTCHA security, reinforcing the necessity for continuous innovation in CAPTCHA design.

Building upon this extensive body of work, our research specifically addresses the design, training, and comparative evaluation of Transformer-enhanced OCR models against traditional methods, emphasizing both performance improvements and security implications for modern CAPTCHA systems.

III. METHODOLOGY

This section outlines the methodology for training a CAPTCHA recognition model using Transformer-based architectures. It includes dataset preparation, data augmentation, model architecture, training procedure, and evaluation metrics. The Experiments are conducted on an NVIDIA GPU RTX 3070.

A. Dataset Preparation

The dataset comprises both real-world CAPTCHAs scraped from various online services and synthetic CAPTCHAs generated using Python's CAPTCHA library. The dataset contains CAPTCHA images with text lengths varying from 4 to 6 characters. The character set includes digits (0–9) and uppercase letters (A–Z). Dataset is composed of synthetic CAPTCHAs, generated using the captcha Python library. Custom variations include: Background noise, Overlapping characters, Character rotations, Affine transformations, Variable-length sequences (4-6 characters per CAPTCHA)

The Structure of the dataset is in csv format with each CAPTCHA is stored as an image (.png, .jpg format). A corresponding CSV file maps filenames to text labels. Dataset has been splitted into Train, validation and Test set with the following numbers of sample as mentioned in table 3.1.

Set	Number of Samples
Training set	100,000
Validation set	20,000
Test set	10,000

Table 3.1: Dataset split

B. Data Preprocess

To improve model generalization and robustness against adversarial attacks, we apply various augmentation techniques during training.

1) Augmentation Techniques

Random rotation within a range of 15°, gaussian noise injection to simulate real-world distortions, color jittering by adjusting brightness, contrast, and saturation. Performed elastic distortions to simulate stretching effects, also random occlusion to mask parts of the CAPTCHA with random lines or shape

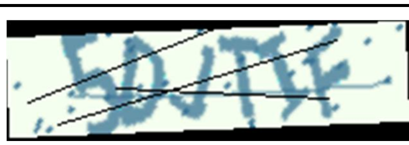
Images	Label
	ZCMQM
	52TWTX
	3J5W
	5DJTIF

Table 3.2: Sample Images

2) Preprocessing Pipeline

a) Convert image to grayscale.
b) Apply augmentation (only during training).
c) Resize image to a fixed dimension (50×200 pixels).
d) Normalize pixel values to [-1, 1].
e) Convert text labels to indexed format.

C. Model Architecture

To compare the effectiveness of Hybrid CNN-Transformer, In this paper two different architectures CRNN and Hybrid CNN-Transforme are going to be discussed.

1) Model Architecture of Baseline CRNN Model

The baseline Convolutional Recurrent Neural Network (CRNN) designed to extract and process sequential features from input images. The architecture comprises three primary modules: a Convolutional Neural Network (CNN) for feature extraction, a feature projection layer, and a Bidirectional Long Short-Term Memory (LSTM) network for sequence modeling.

Stage	Operation / Layer	Parameters / Notes	Output Shape
Input	–	Grayscale image	(B, 1, imgH, W)
CNN Block 1	Conv2d + ReLU	in_channels=1, out_channels=64, kernel_size=3, padding=1	(B, 64, imgH, W)
	MaxPool2d	kernel_size=2, stride=2	(B, 64, imgH/2, W/2)
	Dropout	p=0.3	(B, 64, imgH/2, W/2)
CNN Block 2	Conv2d + ReLU	in_channels=64, out_channels=128, kernel_size=3, padding=1	(B, 128, imgH/2, W/2)
	MaxPool2d	kernel_size=2, stride=2	(B, 128, imgH/4, W/4)
	Dropout	p=0.3	(B, 128, imgH/4, W/4)
CNN Block 3	Conv2d + ReLU	in_channels=128, out_channels=256, kernel_size=3, padding=1	(B, 256, imgH/4, W/4)
	Conv2d + ReLU	in_channels=256, out_channels=256, kernel_size=3, padding=1	(B, 256, imgH/4, W/4)
	MaxPool2d	kernel_size=2, stride=(2, 1)	(B, 256, imgH/8, W/4)
	Dropout	p=0.3	(B, 256, imgH/8, W/4)
Feature Projection	Linear Layer	Transforms vector of size 256×(imgH/8) to a fixed dimension (512)	(B, W/4, 512)
Sequence Modeling	Bidirectional LSTM	input_size=512, hidden_size=128, num_layers=2, dropout=0.3 (results in 256 per time step)	(B, W/4, 256)
Classification	Linear Layer	Maps LSTM output from 256 to num_classes	(B, W/4, num_classes)
Activation	Log Softmax	Applied over the class dimension	(W/4, B, num_classes) (after permuting axes)

Table 3.3: Model Architecture of Baseline CRNN Model

Here, B denotes the batch size. The output shape after the CNN blocks assumes that width is similarly reduced. W/4 reflects the width reduction from the pooling operations. The final log softmax is computed after permuting the dimensions to match the expected format for sequence processing.

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue XI Nov 2025- Available at www.ijraset.com

2) *Model Architecture of Hybrid CNN-Transformer*

This Hybrid CNN-Transformer consists of three main components:

a) Feature Extractor (CNN Backbone) – Extracts spatial features from CAPTCHA images. CNN reduces spatial dimensionality, making the Transformer more efficient and computationally feasible. CNN Feature Extractor efficiently captures local patterns, such as character edges and distortions.

b) Sequence Model (Transformer Encoder) – Processes extracted features to capture long-range dependencies between characters. Transformer Encoder helps model global dependencies between characters.

c) Character Decoder (Fully Connected & CTC) – Maps the processed features to predicted characters.

Stage	Operation / Layer	Parameters / Notes	Output Shape	Input Shape
Input	–	Grayscale image	(B, 1, imgH, W)	–
CNN Block 1	Conv2d + ReLU	in_channels=1, out_channels=64, kernel_size=3, padding=1	(B, 64, imgH, W)	(B, 1, imgH, W)
	MaxPool2d + Dropout	kernel_size=2, stride=2, p=dropout	(B, 64, imgH/2, W/2)	(B, 64, imgH, W)
CNN Block 2	Conv2d + ReLU	in_channels=64, out_channels=128, kernel_size=3, padding=1	(B, 128, imgH/2, W/2)	(B, 64, imgH/2, W/2)
	MaxPool2d + Dropout	kernel_size=2, stride=2, p=dropout	(B, 128, imgH/4, W/4)	(B, 128, imgH/2, W/2)
CNN Block 3	Conv2d + ReLU	in_channels=128, out_channels=256, kernel_size=3, padding=1	(B, 256, imgH/4, W/4)	(B, 128, imgH/4, W/4)
	Conv2d + ReLU	in_channels=256, out_channels=256, kernel_size=3, padding=1	(B, 256, imgH/4, W/4)	(B, 256, imgH/4, W/4)
	MaxPool2d + Dropout	kernel_size=2, stride=(2, 1), p=dropout	(B, 256, imgH/8, W/4)	(B, 256, imgH/4, W/4)
Feature Projection	Linear	in_features=256*6, out_features=d_model	(B, W, d_model)	(B, 256, imgH/8, W/4)
Positional Encoding	PositionalEncoding	d_model=d_model	(B, W, d_model)	(B, W, d_model)
Transformer Encoder	TransformerEncoderLayer x num_layers	d_model=d_model, nhead=num_heads,	(B, W, d_model)	(B, W, d_model)

		dropout=dropout		
	TransformerEncoder	num_layers=num_layers, consisting of multiple TransformerEncoderLayers	(B, W, d_model)	(B, W, d_model)
Normalization	LayerNorm	d_model=d_model	(B, W, d_model)	(B, W, d_model)
Classification	Linear	in_features=d_model, out_features=num_classes	(B, W, num_classes)	(B, W, d_model)
Output	LogSoftmax	dim=2	(B, W, num_classes)	(B, W, num_classes)

Table 3.4: Model Architecture of Hybrid CNN-Transformer

Here,
Input Shape: (B, 1, imgH, W) represents the input batch where B is the batch size, 1 is the single channel (grayscale), and imgH and W are the height and width of the input image.
Output Shape: The final output shape is (B, W, num_classes), where B is the batch size, W is the sequence length (width of the image), and num_classes is the number of output classes.

D. Evaluation Metrics

To rigorously assess the performance of our CAPTCHA recognition models, we employ several quantitative metrics that capture both the model's sequence prediction accuracy and its character-level performance. In particular, we utilize the following:

1) Connectionist Temporal Classification (CTC) Loss

Since the alignment between input image sequences and target text is unknown, the models are trained using the Connectionist Temporal Classification (CTC) loss.

$$L_{ctc} = -ln \sum_{\pi \epsilon B^{-1}(y)} \prod_{t=1}^{T} p(\pi_t|x)$$

where x represents the network input, y the target label sequence, π a valid alignment path, T the length of the network output sequence, and B is a many-to-one mapping that removes repeated labels and blanks. Minimizing L_{ctc}encourages the model to assign high probability to alignments that accurately map to the ground truth CAPTCHA texts.

2) Accuracy

The overall recognition accuracy is computed by comparing the predicted text $\hat{y}$ with the ground truth label y for each CAPTCHA. Formally, the accuracy is defined as

$$Accuracy = \frac{1}{N}\sum_{i=1}^{N} 1(\hat{y}_i = y_i)$$

where N is the number of samples and 1(·) is the indicator function which returns 1 if the prediction is correct and 0 otherwise.

3) Character Error Rate (CER)

In addition to overall accuracy, we compute the Character Error Rate (CER) to evaluate the performance at a granular, character-by-character level. The CER is computed as

$$CER = \frac{1}{\sum_{i=1}^{N} |y_i|}\sum_{i=1}^{N} d(\hat{y}_i, y_i)$$

where $d(\hat{y}_i, y_i)$ represents the Levenshtein (edit) distance between the predicted and ground truth character sequences for sample , i and $|y_i|$ is the length of the ground truth sequence. Lower CER values indicate more accurate character-level predictions.

4) Precision, Recall, and F1-Score

For a comprehensive performance analysis, especially in cases where the cost of misrecognition may vary across characters, we calculate precision, recall, and the F1-score at the character level:

Precision (P):

$$P = \frac{True\ Positives}{True\ Positives + False\ Positives}$$

Recall (R):

$$R = \frac{True\ Positives}{True\ Positives + False\ Negatives}$$

F1-Score:

$$F1 = 2 \times \frac{P \times R}{P + R}$$

These metrics are computed by treating each character prediction as a classification decision. Averaging across all characters and samples yields an overall measure of the model's discriminative performance.

5) Qualitative Analysis via Visualization

To complement the quantitative metrics, we perform qualitative analysis by overlaying the predicted CAPTCHA text on the corresponding images. This visual inspection facilitates the identification of systematic errors such as misclassification due to occlusions or distortions, providing insights for potential improvements.

E. Hyperparameters

Hyperparameter	Value
Learning Rate	0.0001
Optimizer	AdamW
Batch Size	64
Epochs	50
Dropout Rate	0.3
Attention Heads	8
Model Dimensions	224
Attention Layer	4

Table 3.5: Hyperparameters of Hybrid CNN-Transformer Trained Model

IV. RESULTS AND SECURITY ANALYSIS

In this section, we present and analyze the experimental results obtained from the evaluation of the proposed models. We compare the performance of the baseline CRNN model with the Hybrid CNN-Transformer architecture using several quantitative metrics, including overall accuracy, Character Error Rate (CER), precision, recall, and F1-score.

A. Quantitative Performance Comparison

Table 4.1 summarizes the key performance metrics for the CRNN and Hybrid CNN-Transformer models. These metrics are computed on the test set, and the results are averaged over multiple runs to ensure statistical significance.

Metric	CRNN	Hybrid CNN-Transformer
Accuracy	79.3%	95.1%
CER	0.0628	0.0116
Precision	0.9406	0.9887
Recall	0.9848	0.9886
F1-Score	0.9622	0.9887

Table 4.1: Comparison of performance metrics between the baseline CRNN and the Hybrid CNN-Transformer models on Test dataset.

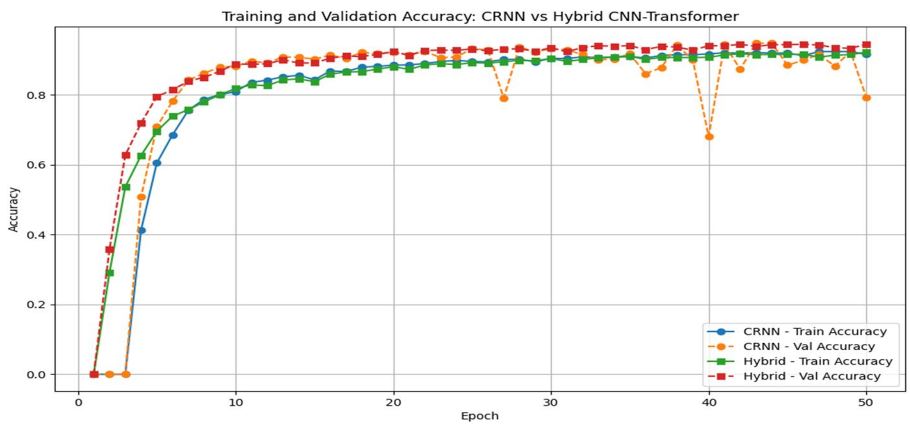


Figure 4.1: Training and validation Accuracy curves for CRNN and Hybrid CNN-Transformer models

The results indicate that the Hybrid CNN-Transformer outperforms the baseline CRNN model across all metrics. In particular, the lower CER value suggests that the Transformer-based architecture is more effective in capturing the long-range dependencies within the CAPTCHA sequences, resulting in fewer character-level errors.

B. Training Convergence and Loss Behavior

Figure 4.2 illustrates the training and validation loss curves over the course of 50 epochs for both models. The faster convergence and lower plateau of the validation loss in the Hybrid CNN-Transformer model indicate that the model not only fits the training data effectively but also generalizes better to unseen samples. The optimization process minimizes the CTC loss.

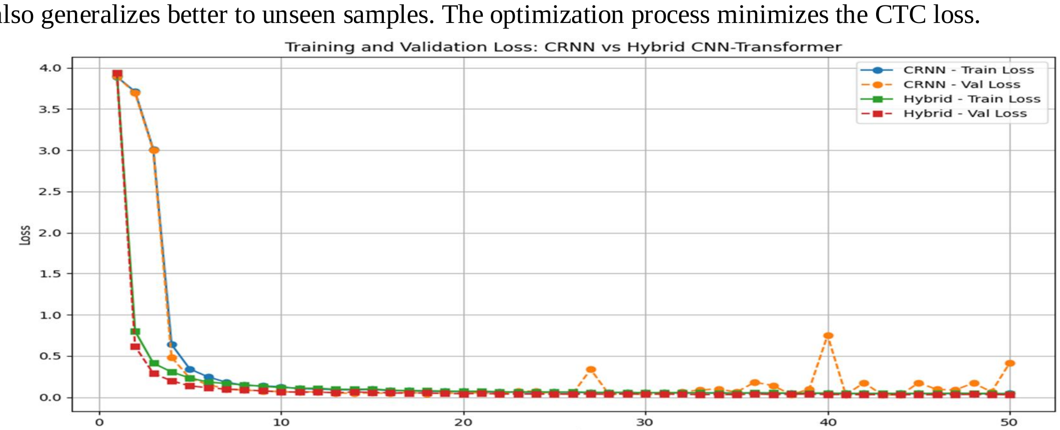


Figure 4.2: Training and validation loss curves for CRNN and Hybrid CNN-Transformer models

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue XI Nov 2025- Available at www.ijraset.com

C. *Qualitative Analysis*

To complement the quantitative results, qualitative analysis was performed by overlaying the predicted CAPTCHA text on sample images from the validation set. Visual inspection reveals that the Hybrid CNN-Transformer model exhibits a higher robustness to distortions and occlusions. For instance, in challenging cases with significant background noise or overlapping characters, the Transformer-based approach more consistently produces the correct sequence compared to the CRNN model.

Figure 4.3 shows representative examples where the Hybrid CNN-Transformer correctly recognized distorted CAPTCHAs, whereas the CRNN occasionally misclassified one or more characters. Such observations underscore the benefit of incorporating global context through self-attention mechanisms inherent to the Transformer.

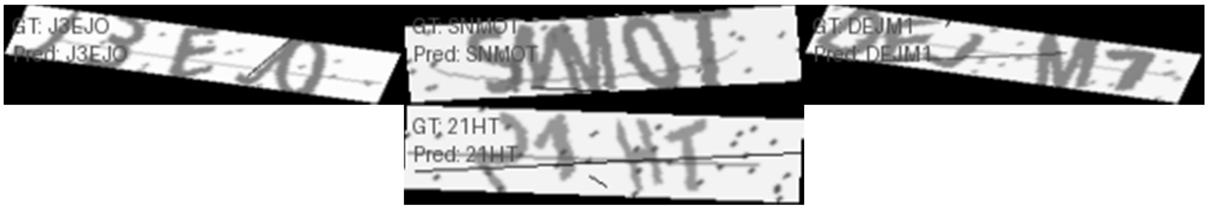


Figure 4.3: Sample visualizations of Hybrid CNN-Transformer model predictions on CAPTCHAs

D. *Security Analysis*

The experimental results indicate that Transformer-based models, particularly the hybrid CNN-Transformer approach, significantly outperform traditional CNN-based OCR systems in terms of CER and WER. The attention mechanism in Transformer layers effectively captures global dependencies, improving recognition accuracy on complex CAPTCHA images.

The improved performance of Transformer-based OCR models raises significant security implications. As deep learning techniques evolve, CAPTCHAs that once relied on distortions and noise may no longer be sufficient to thwart automated attacks. Our work demonstrates that adversaries could leverage these advanced models to bypass CAPTCHA systems, thereby compromising the security of web services. This highlights the urgent need for designing more resilient CAPTCHA schemes that incorporate dynamic and multi-modal elements.

E. *Security Implications & Countermeasures*

Given the vulnerability of current CAPTCHA systems to advanced deep learning-based solvers, we propose the following countermeasures:

1) Adversarial CAPTCHA Generation: Utilize Generative Adversarial Networks (GANs) to produce dynamic CAPTCHAs that adapt to known OCR weaknesses.
2) Multi-Modal Authentication: Combine text, image, and behavioral biometric data to reinforce CAPTCHA security.
3) Time-Based Solving Restrictions: Implement monitoring of CAPTCHA solving times to detect anomalous patterns indicative of automated bots.

These countermeasures, alongside continued research into adversarial training, may offer a robust defense against the evolving threat landscape.

V. CONCLUSION & FUTURE WORK

In this paper, we presented Transformer-based OCR models for CAPTCHA recognition and demonstrated that hybrid CNN-Transformer models outperform traditional methods. While these models achieve lower error rates, their success also emphasizes the growing vulnerability of CAPTCHA systems. Future work will involve:

1) Conducting extensive ablation studies to further optimize model architecture.
2) Integrating adversarial training methods to improve model robustness.
3) Exploring multi-modal CAPTCHA systems that incorporate additional security layers.
4) Performing large-scale experiments on diverse CAPTCHA datasets.

Our findings underscore the need for more sophisticated CAPTCHA designs to keep pace with advances in deep learning.

Declaration of generative AI and AI-assisted technologies in the writing process

During the preparation of this work the author(s) used ChatGPT and QuillBot in order to Ai-assisted writing for better reader's understanding and grammar check . After using this tool/service, the author(s) reviewed and edited the content as needed and take(s) full responsibility for the content of the publication.

REFERENCES

[1] Von Ahn, L., Blum, M., Hopper, N.J. and Langford, J., 2003, May. CAPTCHA: Using hard AI problems for security. In International conference on the theory and applications of cryptographic techniques (pp. 294-311). Berlin, Heidelberg: Springer Berlin Heidelberg.
[2] Goodfellow, I.J., Bulatov, Y., Ibarz, J., Arnoud, S. and Shet, V., 2013. Multi-digit number recognition from street view imagery using deep convolutional neural networks. arXiv preprint arXiv:1312.6082.
[3] Jaderberg, M., Simonyan, K., Vedaldi, A. and Zisserman, A., 2016. Reading text in the wild with convolutional neural networks. International journal of computer vision, 116, pp.1-20.
[4] Hochreiter, S. and Schmidhuber, J., 1997. Long short-term memory. Neural computation, 9(8), pp.1735-1780.
[5] Ranzato, M.A., Chopra, S., Auli, M. and Zaremba, W., 2015. Sequence level training with recurrent neural networks. arXiv preprint arXiv:1511.06732.
[6] Lipton, Z.C., Berkowitz, J. and Elkan, C., 2015. A critical review of recurrent neural networks for sequence learning. arXiv preprint arXiv:1506.00019.
[7] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł. and Polosukhin, I., 2017. Attention is all you need. Advances in neural information processing systems, 30.
[8] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S. and Uszkoreit, J., 2020. An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929.
[9] Kong, S. and Chang, Y.C., 2023, May. A CAPTCHA Recognition Algorithm Based on Character Segmentation and Random Forest. In Journal of Physics: Conference Series (Vol. 2504, No. 1, p. 012036). IOP Publishing.
[10] Noury, Z. and Rezaei, M., 2020. Deep-CAPTCHA: a deep learning based CAPTCHA solver for vulnerability assessment. arXiv preprint arXiv:2006.08296.
[11] Kumar, M., Jindal, M.K. and Kumar, M., 2022. A systematic survey on CAPTCHA recognition: types, creation and breaking techniques. Archives of Computational Methods in Engineering, 29(2), pp.1107-1136.
[12] LeCun, Y., Boser, B., Denker, J.S., Henderson, D., Howard, R.E., Hubbard, W. and Jackel, L.D., 1989. Backpropagation applied to handwritten zip code recognition. Neural computation, 1(4), pp.541-551.
[13] Challagundla, B.C., Gogireddy, Y.R. and Peddavenkatagari, C.R., 2024. Efficient CAPTCHA image recognition using convolutional neural networks and long short-term memory networks. International Journal of Scientific Research in Engineering and Management (IJSREM), 8(3), pp.1-5.
[14] Papa, L., Russo, P., Amerini, I. and Zhou, L., 2024. A survey on efficient vision transformers: algorithms, techniques, and performance benchmarking. IEEE Transactions on Pattern Analysis and Machine Intelligence.
[15] Wan, X., Johari, J. and Ruslan, F.A., 2024. Adaptive-CAPTCHA: A Text CAPTCHA Solver based on CRNN and Configurable Filter Networks.
[16] Li, X., Ding, H., Yuan, H., Zhang, W., Pang, J., Cheng, G., Chen, K., Liu, Z. and Loy, C.C., 2024. Transformer-based visual segmentation: A survey. IEEE transactions on pattern analysis and machine intelligence.
[17] Derea, Z., Zou, B., Al-Shargabi, A.A., Thobhani, A. and Abdussalam, A., 2023. Deep Learning Based CAPTCHA Recognition Network with Grouping Strategy. Sensors, 23(23), p.9487.
[18] Kumar, M., Jindal, M.K. and Kumar, M., 2022. A systematic survey on CAPTCHA recognition: types, creation and breaking techniques. Archives of Computational Methods in Engineering, 29(2), pp.1107-1136.
[19] Igbekele, E.O., Adebiyi, A.A., Ibikunle, F.A., Adebiyi, M.O. and Oludayo, O.O., 2021. Research trends on CAPTCHA: A systematic literature. International Journal of Electrical and Computer Engineering, 11(5), p.4300.
[20] Kanoosh, H.M., Abbas, A.F., Kamal, N.N., Khadim, Z.M., Majeed, D.A. and Algburi, S., 2024, May. Image-Based CAPTCHA Recognition Using Deep Learning Models. In Proceedings of the Cognitive Models and Artificial Intelligence Conference (pp. 273-278).
[21] Puneet and Deepika, 2024, March. Redefining Security: Unveiling the Vulnerabilities of Captcha Mechanisms Using Deep Learning. In 2024 International Conference on Emerging Smart Computing and Informatics (ESCI) (pp. 1-6). IEEE.
[22] Shi, C., Xu, X., Ji, S., Bu, K., Chen, J., Beyah, R. and Wang, T., 2021. Adversarial captchas. IEEE transactions on cybernetics, 52(7), pp.6095-6108.
[23] Li, C., Chen, X., Wang, H., Wang, P., Zhang, Y. and Wang, W., 2021. End-to-end attack on text-based CAPTCHAs based on cycle-consistent generative adversarial network. Neurocomputing, 433, pp.223-236.