CAPTCHAs recognition based on the central region of the convolutional feature map

Viet Tran Quoc
Department of Information Technology
FPT University
Ho Chi Minh, Vietnam
viettqse150044@fpt.edu.vn

Duy Nguyen
Department of Information Technology
FPT University
Ho Chi Minh, Vietnam
duynse150736@fpt.edu.vn

Trung Nguyen Quoc
Department of Information Technology
FPT University
Ho Chi Minh, Vietnam
trungng46@fpt.edu.vn

Vinh Truong Hoang
Faculty of Information Technology
Ho Chi Minh City Open University, Vietnam
vinh.th@ou.edu.vn

Tuan Le-Viet
Faculty of Information Technology
Ho Chi Minh City Open University, Vietnam
tuan.lv@ou.edu.vn

***Abstract*—CAPTCHA (Completely Automated Public Turing Test to Tell Computers and Humans Apart) is a crucial human-machine distinction tool that websites employ to thwart automated malicious program attacks. Investigating CAPTCHA recognition can reveal weaknesses in CAPTCHA systems. By leveraging deep learning and computer vision techniques, the very purpose of CAPTCHAs can be circumvented. A Deep Convolutional Neural Network model is employed to identify CAPTCHAs, eliminating the need for traditional image processing techniques such as location and segmentation. Our research proposes a CAPTCHA recognition system focusing on the central area of feature maps using the DCNN model, which we customized and combined with the attention mechanism. This approach helps distill the character information that needs to be learned during training in the complex context of CAPTCHAs with lots of noise. The experimental findings illustrate that our model has exceptional identification capabilities on CAPTCHAs that contain background noise and character adhesion distortion. It achieves excellent accuracy and a low character mistake rate across several datasets.**

***Index Terms*—CAPTCHA recognition, deep convolutional neural network, security, deep learning, attention mechanism**

I. Introduction

In today's digital age, where the internet plays an integral role in our lives, the need for security and authentication has never been more critical. The advent of CAPTCHA technology has ushered in a new era of online security, addressing the ever-present threat of automated bots and malicious software infiltrating web services. This technology has become a cornerstone of online security, serving as the first line of defense against automated scripts and bots attempting to gain unauthorized access to web services. CAPTCHAs are ubiquitous in our online experiences. From signing up for a new email account to purchasing online, CAPTCHAs are the first line of defense against automated scripts and bots attempting to gain unauthorized access. The fundamental concept of CAPTCHA technology revolves around distinguishing between humans and machines, which should be a straightforward task for humans but a significant challenge for automated scripts. While CAPTCHA technology has proved invaluable in safeguarding online platforms, it presents a conundrum for automated systems designed to access web services legitimately, such as web crawlers and data mining tools. This friction between security and legitimate access underscores the critical need for CAPTCHA recognition models. These models must be capable of deciphering and solving CAPTCHAs, effectively bridging the gap between automated systems and web services. Developing a reliable CAPTCHA recognition model is not merely a matter of convenience. It is a fundamental requirement for several reasons:

A. Real-World Application of CAPTCHA Recognition Models

- Enhancing User Experience: despite their security benefits, CAPTCHAs are often perceived as a necessary inconvenience by users. The need to recognize and solve CAPTCHAs can be time-consuming and frustrating. CAPTCHA recognition models solve this problem by automating the process, resulting in a smoother and more efficient user experience.

- Efficient Web Crawling and Data Retrieval: Legitimate automated systems, like web crawlers used by search engines or data mining tools, require access to web services for various purposes, including indexing content and collecting data. CAPTCHA recognition models are pivotal in facilitating these activities, ensuring that automated systems can access the required information without disruption.
- Market Research and Data Quality: In market research, CAPTCHAs can be used to verify survey respondents' authenticity, ensuring the collected data's quality and integrity. CAPTCHA recognition models can automate this verification process, saving time and resources.

B. The Role of CAPTCHA Recognition in Research

Beyond its real-world applications, CAPTCHA recognition models are a captivating study area within Computer Vision. These models require a deep understanding of image recognition, pattern analysis, and machine learning. Here's why CAPTCHA recognition is of significant interest in Computer Vision research:

- Complex Image Recognition: CAPTCHAs come in various forms, including distorted characters, image puzzles, and more. Recognizing and solving CAPTCHAs demand advanced image recognition techniques and algorithms, making them a compelling challenge for researchers in Computer Vision.
- Pattern Analysis and Machine Learning: CAPTCHA recognition models often rely on machine learning and pattern recognition algorithms. Developing and improving these models allows researchers to explore cutting-edge techniques and advance the state of the art in Computer Vision.
- Practical Application of Computer Vision: CAPTCHA recognition models provide a practical application of Computer Vision in real-world scenarios. This practicality showcases the potential of Computer Vision technologies beyond academic research and highlights the impact of these technologies on the security and efficiency of online services.

In short, this introduction sets the stage for exploring the CAPTCHA recognition model in the context of Computer Vision research. We will delve deeper into the challenges and methodology of CAPTCHA recognition, aiming to develop models that contribute to the evolving landscape of image recognition and security in the digital age. This research endeavors to bridge the gap between humans and machines, making the online world safer and more efficient.

II. RELATED WORKS

Text-based CAPTCHAs have stood as the predominant system in use. Developers of CAPTCHAs have sought to bolster security by integrating diverse resistance mechanisms into established text CAPTCHAs, such as introducing noise, incorporating backdrops, and crowding characters together (referred to as CCT). Nonetheless, research studies [1]–[4] indicate that these resistance mechanisms lack effectiveness. Several research endeavors are dedicated to formulating attack strategies against prevailing text-based CAPTCHAs. Initial efforts to breach different CAPTCHAs involved exploiting estimating distortions, shape characteristics, and employing machine-learning techniques.

Numerous algorithms have been proposed to break down text-based CAPTCHAs into individual characters. Zhang et al. [5] introduced the utilization of the vertical projection technique for CAPTCHA segmentation. They enhanced this approach to address the challenge of fused or conglutinated characters by incorporating size-related character features and their positions into the vertical projection histogram. Additionally, their work encompassed the segmentation of various forms of conglutination. K. Chellapilla and P. Y. Simard [6] adopted the connected component algorithm to segment a variety of CAPTCHA formats, such as those used by Yahoo and Google. Their efforts yield a success rate of up to 66%. It is worth noting that both the vertical projection and connected component algorithms entail multiple preprocessing steps, making them computationally intensive and time-consuming. An alternative CAPTCHA segmentation approach, introduced by Hussain et al. [7], revolves around a recognition-based method for segmentation.

In recent years, significant efforts have been dedicated to scene text recognition. To gain a comprehensive understanding of text recognition, interested readers can consult the recent survey by Ye and Doermann [8]. Conventionally, two primary categories of approaches exist: the bottom-up and top-down methods.

The early approaches primarily focused on employing bottom-up techniques. These techniques involved initially detecting individual characters one by one through methods such as sliding windows [9], [10], connected components [11], or Hough voting [12]. Subsequently, these detected characters were integrated to form the output text.

Conversely, the top-down approaches follow a different paradigm. They aim to predict the entire text directly from the original image without character-level detection. An example is the work by Jaderberg et al., who devised a convolutional neural network (CNN) with a structured output layer for unconstrained text recognition [13]. They also undertook a challenging task involving classifying 90,000 English words using a CNN, where each class represents a distinct word [14].

Recent advancements in this field have cast the problem as a sequence recognition task, where images and texts are encoded separately as sequences of patches and characters. Sutskever et al. [15] leveraged sequences of HOG features to represent images and generated character sequences using recurrent neural networks (RNN). Similarly, Shi et al. [16] introduced an end-to-end neural network that combines CNN and RNN. They also pioneered the development of an attention-based spatial transformer network (STN) for

rectifying text distortion, which greatly aids in recognizing curved scene texts [17].

Unlike the abovementioned approach, in this paper, we extract deeper representations of images using a ResNet-based CNN. This may be the first work on scene text recognition using CNN based on ResNet – Residual Network backbone (with 32 layers that we customized) and applying crop function to directly select the central area to focus on. Then, we feed the sequence of features to the Attention module so that the model learns the context of the image's important feature associations and can thus decode the text with high accuracy. At the same time, we re-implement methods such as CRNN and vision transformer on the datasets and compare the three approaches.

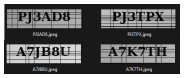
III. DATASET

A. Data Collection

We simultaneously use support libraries to create datasets and collect data from the internet. As a result, we have three large and complex datasets. We have labeled all data during the preparation. Then, we began the process of Exploratory data analysis (EDA).

Exploratory Data Analysis (EDA) is an important step in the data exploration before model building begins. This step helps us better understand the data, the characteristics, and the challenges of each dataset. Below is the statistical table 1 we obtained after performing this process:

TABLE I: CAPTCHAs dataset.

Dataset	Description	Sample
1	• The number of characters: 5 characters. • The number of images: 113,062 images. • Size: 150×40. • Challenge: Color diversity: background and text. Lots of colorful horizontal noise.	
2	• The number of characters: 5 characters. • The number of images: 103,250 images. • Size: 200×100. • Challenge: Color speckle noise and horizontal stripe. Pattern black noise - black horizontal stripes are added. Diversity of colors between text in images	
3	• The number of characters: 6 characters. • The number of images: 26,255 images. • Size: 198×50. • Challenge: Grid (horizontal stripes and vertical stripes). Contrast varies between images.	

B. Data Preprocessing

The process is divided into 3 parts as follows:

- Before Batch Transform (Resize): This is a pre-batch transform that is used to ensure that all images in the batch have the same size. Use padding mode to ensure the correct characteristics are retained when resizing the image. This transformation will perform the following steps:
  - Read each image in the batch.
  - Resize the image to a fixed size.
  - This transform will preserve the aspect ratio of the image if the condition is satisfied.
  - Convert the image to a tensor.
- Create Batch Transform: This is a transform that is used to create a batch of images from single images. This transformation will perform the following steps:
  - Create an empty tensor to store the batch of images.
  - Add each image that has been transformed previously to the batch tensor.
  - Return the batch tensor.
- After Batch Transform: This is a list of post-batch transforms that are used to convert the images to tensors and normalize them. This transformation will perform the following steps:
  - Convert each image from the integer (int) data type to the floating-point (float) data type.
  - Return the converted image, then normalize the images based on the provided mean and variance. In this case, the mean and variance of the RGB channels are both 0.5.
  - Return the normalized images.

IV. THE PROPOSED METHOD

During our research, we found that only some information condensed through the extraction of convolutional layers yields the target for each prediction label. Additionally, it is susceptible to the influence of complex or low-quality scene data, leading to the generation of inaccurate alignment factors and potentially causing a mismatch between attention regions and the actual ground-truth regions. To solve this problem, we propose a two-step approach that determines the attention center for each predicted label and directs attention to the target regions. To do this, we first leverage residual blocks with skip connections presented by He et al. [18]. We modified the Resnet-based CNN architecture as follows:

The ResNet32 layers-based Convolutional Neural Network (CNN) architecture includes building blocks, ResNet blocks in bold, and output size. In there, the convolution layers have the following format: $kernel_W \times kernel_H, stride_W \times stride_H, pad_W \times pad_H, channels$, the maxpooling layers have the following format: $kernel_W \times kernel_H, stride_W \times stride_H, pad_W \times pad_H$, and the ResNet blocks have the

TABLE II: A RESNET-BASED CNN ARCHITECTURE

Stage	Description	Output size
Stage 1	3 x 3, 1 x 1, 1 x 1,32	32 x 256
	3 x 3, 1 x 1, 1 x 1,32	
Stage 2	maxpool: 2 × 2, 2 × 2, 0 × 0	
	3 × 3, 128 **3 × 3, 128** **(1 ResNet blocks)**	16 x 128
	3 × 3, 1 × 1, 1 × 1, 128	
Stage 3	maxpool: 2 × 2, 2 × 2, 0 × 0	8 x 64
	3 × 3, 256 **3 × 3, 256** **(2 ResNet blocks)**	
	3 × 3, 1 × 1, 1 × 1, 256	
Stage 4	maxpool: 2 × 2, 1 × 2, 1 × 0	4 x 65
	3 × 3, 512 **3 × 3, 512** **(5 ResNet blocks)**	
	3 × 3, 1 × 1, 1 × 1, 512	
Stage 5	**3 × 3, 512** **3 × 3, 512** **(3 ResNet blocks)**	1 x 65
	2 × 2, 1 × 2, 1 × 0, 512	
	2 × 2, 1 × 1, 0 × 0, 512	

following format: kernel size, number of channels. The symbols W and H denote the width and height of feature maps.

During the t-th step, we proceed with the extraction of a feature map patch of dimensions $P(P_H, P_W)$ from a convolution output in the following manner:

$$M_t = Crop(M, P_H, P_W) \tag{1}$$

where M is the convolution feature map P is set to the max-size of ground-truth regions.

The attention mechanism of attention-based decoding architecture was originally introduced by Bahdanau et al. [19]. However, instead of using the attention mechanism to focus on the current and previous contexts, we use it to direct attention to target regions by creating probability distributions over the designated attention areas.

Using the extracted feature maps(1), we calculate the energy distribution across the attention region in the following manner:

$$e_t^{(i,j)} = tanh(Vg_t + WM_t^{(i,j)} + b) \tag{2}$$

V, W and b denote trainable parameters, (i, j) refers to the $(iP_W + j)$-th feature vector, and g_t represents the result of the summation of feature vectors $(h_1, \ldots, h_T)$ in a sequential manner with assigned weights,

$$g_t = \sum_{j=1}^{T} \alpha_{tj} h_j \tag{3}$$

where $\alpha_t \in \mathbb{R}^T$ is a vector of attention weights, $_t$ is often evaluated by scoring individual elements within the set $(h_1, \ldots, h_T)$ separately. Subsequently, the probability distribution across the chosen region is calculated as follows:

$$P_t^{(i,j,k)} = \frac{exp(e_t^{(i,j,k)})}{\sum_{k'}^{K} exp(e_t^{(i,j,k')})} \tag{4}$$

where K is the number of label classes.

Then, the loss function is:

$$L = -\sum_{t}^{N} \sum_{i}^{P_W} \sum_{j}^{P_H} logP(\hat{y}_t^{(i,j)} | \tau, \omega) \tag{5}$$

where $\hat{y}_t^{(i,j)}$ is the ground-truth pixel label, τ is an input image and ω is a vector that combines all parameters. The loss is added only for the subset of images with character annotations.

V. EXPERIMENTS AND RESULTS

A. Evaluation metrics

Character Error Rate (CER) measures the similarity between two strings by counting the minimum number of character-level operations (insertions, deletions, or substitutions) required to transform one string into the other.

$$CER = \frac{S + D + I}{N} \tag{6}$$

where S: number of substitutions; D: number of deletions; I: number of insertions; N: total number of characters in the reference text $(N = S + D + C)$; C: number of correct characters.

Accuracy: a measure of how well the model predicts the correct output compared to the total number of predictions. Accuracy is often expressed as a percentage. The accuracy of a model is calculated using the following formula:

$$ACC = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}} \tag{7}$$

B. Experience setup

All experiments were run on a workstation equipped with an *Intel(R) Xeon(R) Gold 5320 CPU @ 2.20 GHz with 256 GB of RAM, and an NVIDIA A10 GPU with 24 GB of RAM*.

C. Baseline model

To facilitate comparative analysis, we opted for two contemporary state-of-the-art models as reference points: CRNN [16] with a ResNet50 backbone and the vision transformer (ViT) [20]. CRNN, recognized for its previous achievements in recognition tasks, stands as a reliable benchmark. Notably, ViT signifies the latest progress in the realm of text recognition, as indicated by its noteworthy performance metrics documented in recent literature.

D. Model training

For the optimization process, we use ADADELTA [21], which dynamically computes learning rates for each dimension. ADADELTA is a variant of AdaGrad, distinguishing itself by reducing the extent to which the learning rate varies across coordinates. Notably, ADADELTA is often recognized as an algorithm that operates without explicitly using a learning rate, as it adapts based on the current change to calibrate future adjustments. In contrast to the traditional momentum method [22], ADADELTA eliminates the need to set a learning rate manually. Crucially, our observations indicate that optimization with ADADELTA leads to faster convergence compared to the momentum method.

E. Resutls

Our model demonstrates exceptional performance across all three datasets, outperforming the baseline models in terms of both accuracy (ACC) and character error rate (CER) according to the evaluation metrics.

TABLE III: RESULTS

Models	Metrics	Dataset 1	Dataset 2	Dataset 3
CRNN (ResNet50 backbone)	ACC	0.6585	0.4329	0.9729
	CER	0.0956	0.1232	0.0039
ViT	ACC	0.87132	0.9067	0.9988
	CER	0.0244	0.01411	0.00016
Ours	**ACC**	**0.9421**	**0.96350**	**0.9998**
	CER	**0.0106**	**0.0053**	**0.0000272**

F. Discussion

In the discussion, we thoroughly compare our model with existing research, specifically emphasizing its advances over models like ViT in the domain of text-based CAPTCHA recognition. We underscore that while ViT and equivalent models have established the groundwork for recognizing CAPTCHA characters in specific datasets, our model excels in managing noisy datasets and showcases heightened complexity in the recognition context. This detailed analysis underscores the technical superiority of our attention-focused combined target-region approach, situating them within the broader landscape of CAPTCHA recognition research. Besides, the discussion admits the current limitations in the practical application scope of our model and proposes expanding future research to encompass diverse datasets and explore feasible integration. This strategy not only responds to feedback but also explains how our model not only builds upon but also surpasses existing methods in the field.

VI. CONCLUSIONS

In summary, our approach emerges as a formidable choice for those prioritizing recognition models with remarkable accuracy and minimal character error rates. Using multiple datasets with various characters and image sizes and the challenges of high complexity and noise makes evaluating our model even more valuable. Future endeavors in this domain should delve deeper into the specific factors influencing the real application, paving the way for enhanced efficiency in model development and deployment.

REFERENCES

[1] H. Gao, M. Tang, Y. Liu, P. Zhang, and X. Liu, "Research on the security of microsoft's two-layer captcha," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 7, pp. 1671–1685, 2017.

[2] H. Gao, J. Yan, F. Cao, Z. Zhang, L. Lei, M. Tang, P. Zhang, X. Zhou, X. Wang, and J. Li, "A Simple Generic Attack on Text Captchas," in *Proceedings 2016 Network and Distributed System Security Symposium*. San Diego, CA: Internet Society, 2016. [Online]. Available: https://www.ndss-symposium.org/wp-content/uploads/2017/09/simple-generic-attack-text-captchas.pdf

[3] F. C. Z. Z. L. L. J. Q. X. L. Haichang Gao, Xuqin Wang, "Robustness of text-based completely automated public turing test to tell computers and humans apart." Internet Society, 2016.

[4] H. Gao, W. Wang, J. Qi, X. Wang, X. Liu, and J. Yan, "The robustness of hollow captchas," in *Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security*, ser. CCS '13. New York, NY, USA: Association for Computing Machinery, 2013, p. 1075–1086. [Online]. Available: https://doi.org/10.1145/2508859.2516732

[5] L. Zhang, Y. Xie, X. Luan, and J. He, "Captcha automatic segmentation and recognition based on improved vertical projection," in *2017 IEEE 9th International Conference on Communication Software and Networks (ICCSN)*, 2017, pp. 1167–1172.

[6] K. Chellapilla and P. Y. Simard, "Using machine learning to break visual human interaction proofs (hips)," in *Proceedings of the 17th International Conference on Neural Information Processing Systems*, ser. NIPS'04. Cambridge, MA, USA: MIT Press, 2004, p. 265–272.

[7] R. Hussain, H. Gao, and R. A. Shaikh, "Segmentation of connected characters in text-based CAPTCHAs for intelligent character recognition," *Multimedia Tools and Applications*, vol. 76, no. 24, pp. 25 547–25 561, Dec. 2017. [Online]. Available: https://doi.org/10.1007/s11042-016-4151-2

[8] Q. Ye and D. Doermann, "Text Detection and Recognition in Imagery: A Survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 7, pp. 1480–1500, Jul. 2015. [Online]. Available: http://ieeexplore.ieee.org/document/6945320/

[9] K. Wang, B. Babenko, and S. Belongie, "End-to-end scene text recognition," in *2011 International Conference on Computer Vision*, 2011, pp. 1457–1464.

[10] K. Wang and S. Belongie, "Word spotting in the wild," in *Computer Vision – ECCV 2010*, K. Daniilidis, P. Maragos, and N. Paragios, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 591–604.

[11] L. Neumann and J. Matas, "Real-time scene text localization and recognition," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 3538–3545.

[12] C. Yao, X. Bai, B. Shi, and W. Liu, "Strokelets: A learned multi-scale representation for scene text recognition," in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 4042–4049.

[13] M. Jaderberg, K. Simonyan, A. Vedaldi, and A. Zisserman, "Deep Structured Output Learning for Unconstrained Text Recognition," Apr. 2015, arXiv:1412.5903 [cs]. [Online]. Available: http://arxiv.org/abs/1412.5903

[14] ——, "Reading text in the wild with convolutional neural networks," *International journal of computer vision*, vol. 116, pp. 1–20, 2016.

[15] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to Sequence Learning with Neural Networks."

[16] B. Shi, X. Bai, and C. Yao, "An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition," *IEEE Transactions on Pattern Analysis amp; Machine Intelligence*, vol. 39, no. 11, pp. 2298–2304, nov 2017.

[17] B. Shi, X. Wang, P. Lyu, C. Yao, and X. Bai, "Robust scene text recognition with automatic rectification," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 4168–4176.

[18] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[19] D. Bahdanau, K. Cho, and Y. Bengio, "Neural Machine Translation by Jointly Learning to Align and Translate," May 2016, arXiv:1409.0473 [cs, stat]. [Online]. Available: http://arxiv.org/abs/1409.0473

[20] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30. Curran Associates, Inc., 2017.

[21] M. D. Zeiler, "ADADELTA: An Adaptive Learning Rate Method," Dec. 2012, arXiv:1212.5701 [cs]. [Online]. Available: http://arxiv.org/abs/1212.5701

[22] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, *Learning Representations by Back-Propagating Errors*. Cambridge, MA, USA: MIT Press, 1988, p. 696–699.