Proceedings of the 2nd International Conference on Machine Learning and Automation
DOI: 10.54254/2755-2721/81/20240967

Research on CAPTCHA recognition technology based on deep learning

Shengyuan Tang

College of Mathematics and Information & College of Software Engineering, South China Agricultural University, Guangzhou, 510642, China

beary2048@gmail.com

Abstract. CAPTCHA (Completely Automated Public Turing test to tell Computers and Humans Apart), a security technique widely used on the Internet, can be utilized to distinguish human users from automated programs. The rapid development of deep learning technology has led to the emergence of graph-based recognition techniques that demonstrate excellent performance, which has prompted the investigation of CAPTCHA recognition based on deep learning as a research area of significant interest. This paper addresses the challenging problem of CAPTCHA recognition based on deep learning techniques, reviews the development and classification of CAPTCHA, examines traditional CAPTCHA recognition methods, and delves into the application of deep learning in CAPTCHA recognition. Therefore, a CAPTCHA recognition system is designed and its effectiveness is verified through experiments. This paper makes a significant contribution to the field of CAPTCHA recognition by proposing a deep learning-based approach, which not only enhances the accuracy and efficiency of CAPTCHA recognition, but also provides new ideas and methods for the further development. In the future, further research will be conducted in the field of CAPTCHA recognition to explore additional deep learning models and techniques with the aim of boosting the security and user experience of CAPTCHA.

Keywords: CAPTCHA Recognition, Deep Learning, Security Technique, Machine Learning, Image Recognition.

1. Introduction

CAPTCHA, which stands for Completely Automated Public Turing test to tell Computers and Humans Apart, was first conceived in 2000 to distinguish whether a user is a computer or a human [1] [2]. In the contemporary era, CAPTCHA is a prevalent tool employed by numerous websites, largely due to its straightforward maintenance and favorable user experience [3]. However, the rapid development of image recognition technology is posing an increasing challenge to traditional CAPTCHA techniques, making the effective recognition and cracking of CAPTCHAs a significant research topic [4].The paper attempts to develop a deep learning-based CAPTCHA recognition system that aims to ensure recognition accuracy by using a lightweight model, while reducing the recognition time to meet the time constraints of entering CAPTCHAs on websites. The experimental results permit the formulation of suggestions for future CAPTCHA recognition improvements. Therefore, this paper contributes to the improvement of CAPTCHA recognition accuracy and efficiency through the implementation of deep

DOI: 10.54254/2755-2721/81/20240967

learning techniques. The application of advanced deep learning algorithms enables the efficient handling of complex CAPTCHA recognition tasks. Moreover, a comprehensive examination of CAPTCHA recognition technology can reveal its shortcomings, enhance the existing CAPTCHA generation mechanism, reinforce the security of the existing system, and offer novel concepts and methodologies for the future advancement of CAPTCHA technology. The improvement of CAPTCHA recognition and generation techniques in the context of the increasing importance of network security has important practical and application value.

2. Fundamentals of CAPTCHA Recognition

2.1. Overview of CAPTCHA Recognition

CAPTCHA recognition refers to the automated identification and parsing of CAPTCHA content through technical means, allowing computer systems to verify user identity and ensure system security. The main purpose of CAPTCHAs is to prevent automated programs (bots) from abusing network services, such as mass account registrations, voting, and submitting spam. Traditional CAPTCHAs typically include uppercase and lowercase English letters, numbers, and Chinese characters [5]. In addition to the aforementioned character CAPTCHAs and image CAPTCHAs, there are also audio CAPTCHAs [6]. These CAPTCHAs generate random characters or images that users need to recognize and input, thus distinguishing human users from automated programs.

The development of CAPTCHA recognition technology has evolved from simple rule-based methods to complex machine learning-based methods. Early CAPTCHA recognition methods mainly relied on image processing and pattern recognition techniques, using steps such as preprocessing, feature extraction, and classification to recognize CAPTCHAs. However, as CAPTCHA designs became more complex and diverse, these traditional methods revealed limitations such as low recognition accuracy and weak anti-interference capabilities.

In recent years, the rise of deep learning technology has opened up new avenues for CAPTCHA recognition [7]. Deep learning, especially Convolutional Neural Networks (CNNs) [8], demonstrates efficacy in image recognition tasks. Its robust feature extraction capabilities and nonlinear modeling abilities have led to widespread application in CAPTCHA recognition. The construction of a deep learning model enables the automatic learning of complex features in CAPTCHA, thereby enhancing recognition accuracy and robustness. Besides, dynamic CAPTCHAs can be identified using Recurrent Neural Networks (RNN) and Long Short-Term Memory Networks (LSTM), which have advantages in processing continuous input.

2.2. Application of Deep Learning in CAPTCHA Recognition

Deep learning has been widely applied in CAPTCHA recognition due to its powerful feature extraction and automated processing capabilities. CAPTCHA recognition involves complex image processing and pattern recognition tasks, which traditional methods struggle to handle due to the variable nature of CAPTCHA designs. In contrast, deep learning can train high-precision recognition models using large datasets. CNNs are one of the most commonly used models in deep learning, particularly for image recognition tasks. CNNs can automatically extract local and global features from images through stacked convolutional, pooling, and fully connected layers. For CAPTCHA recognition, CNNs can effectively process both character and image CAPTCHAs by extracting features such as edges and shapes of characters through multiple convolutional layers, followed by classification using fully connected layers. Common CNN architectures like DenseNet [9], LeNet, AlexNet, and ResNet [10] have shown excellent performance in CAPTCHA recognition tasks.

DOI: 10.54254/2755-2721/81/20240967

3. Methodology

3.1. Experimental Setup

The experimental setup is divided into two distinct sections: environment configuration and parameter configuration, which aims to ensure that the model is trained and evaluated under consistent and controlled conditions, leading to reproducible and accurate comparison of results.

3.1.1. Environment Configuration. In terms of hardware information, specific hardware settings and environmental configurations were employed in this experiment to ensure optimal performance for the deep learning task. Table 1 provides a detailed overview of the hardware used, including a 4-core CPU, a V100 AI accelerator with 32GB of GPU memory, 32GB of RAM, and 100GB of total storage. Table 2 outlines the software environment, specifying the versions of Python and PaddlePaddle used.

Table 1. Hardware Information for this Experiment

Name	Model or Parameter
CPU	4 cores
AI Accelerator	V100
Total GPU Memory	32GB
Total RAM	32GB
Total Storage	100GB

Table 2. Environment Configuration

Name	Version
Python	3.74
PaddlePaddle	2.3.2

3.1.2. Parameter Configuration. This experiment uses deep learning-based CAPTCHA recognition technology to configure the optimizer, model architecture, loss functions, evaluation metrics, and parameters for training and evaluation. In the global configuration, the total number of training epochs is set to 100, and the training log is printed every 10 batches during training. The pre-training path of the model is set and the cosine learning rate scheduler is used with an initial learning rate of 0.001.

In regard to the model architecture, MobileNetV1Enhance is selected as the backbone network, and a multi-head structure including CTCHead and SARHead is configured. The CTCHead uses SVTR (Sequence to Sequence Visual Text Recognition) as the neck network, configured with 64 dimensions and 2 layers, whereas the SARHead is configured with 512-dimensional encoding and a maximum text length of 25. For the optimizer, the Adam optimizer is utilized for the optimization process, with the beta1 and beta2 parameters set to 0.9 and 0.999, respectively. The loss function uses a multi-loss structure, including CTCLoss and SARLoss, to better adapt to different types of CAPTCHA recognition tasks. Training data is configured with various data augmentation and preprocessing operations, such as image decoding, image enhancement, and image resizing. The batch size for the training dataset is set to 256 per GPU, and the batch size for the evaluation dataset is set to 128 per GPU.

3.2. Dataset Collection and Pre-processing

The dataset used in the experiment is sourced from the AI Studio online community and contains a total of 57,800 CAPTCHA images, all of which have file names corresponding to their contents. To preprocess the CAPTCHA dataset, the `os.walk()` function is used to iterate through all the files and subdirectories within the specified path. For each file, the file extension is verified, and if it is not `.txt` (i.e., it is a non-text file), the filename and its content are added to the result list. Subsequently, the `divide_list()` function is used to split the dataset into training and testing sets according to a specified ratio. The `len()` function is used to get the length of the dataset, and the split index is calculated based

DOI: 10.54254/2755-2721/81/20240967

on the given split ratio. The dataset is then divided into training and testing sets using slicing operations. To generate the character dictionary, the `gen_dict()` function creates a character list that includes numbers and letters to accurately map characters and numbers in subsequent CAPTCHA recognition tasks Finally, the `write_file()` function is used to write the processed data to the specified files, including the training set list file `train_list.txt`, the testing set list file `test_list.txt`, and the character dictionary file `dict.txt`.

3.3. Model Training

The size of the input image is normalized to 48x320 pixels. Prior to the input image being fed into the model for training, several preprocessing steps are performed, including normalization, resizing, and data augmentation. Normalization scales the pixel values to the range [0, 1] to ensure consistent input data distribution. The resizing process scales the image to a specified size. Data augmentation techniques (e.g. rotation, translation, flipping, and color adjustments) are used to increase the diversity of the training dataset and improve the model's generalization ability.

The model architecture adopts MobileNetV1 as the backbone network combined with the SVTR algorithm, including both CTCHead and SARHead, a combination that leverages the lightweight and efficient convolutional layers of MobileNetV1 and the complex sequence recognition capabilities of SVTR. MobileNetV1 is a lightweight CNN designed with fewer parameters and computations. Due to its high efficiency, it is well suited for mobile and embedded devices. Despite its compact size, MobileNetV1 maintains high recognition accuracy, making it an ideal choice for the backbone network. The SVTR algorithm consists of CTCHead and SARHead, where CTCHead is used for Connectionist Temporal Classification to optimize the CTCLoss function, which helps to improve the alignment of the character sequences, and SARHead is utilized for Sequence Attention Based Recognition, which uses the attention mechanism to improve recognition accuracy, and optimize the model using SARLoss function. The model employs MultiLoss, which combines CTCLoss and SARLoss, which ensures that the model optimizes both loss functions during training to improve both character sequence alignment and overall recognition accuracy. The optimizer is selected as Adam, with an initial learning rate set to 0.001. To adapt the learning rate dynamically during training, a cosine annealing learning rate decay strategy is adopted, which gradually reduces the learning rate as training progresses, with the learning rate increasing from a small value to the initial one during the warm-up phase of the first 5 calendar elements. This helps to stabilize the training process and avoid fluctuations in the early stages.

To ensure that multiple versions of the model are available for selection, the model is saved every 3 epochs during training, which facilitates subsequent evaluation and adjustment to select the best performing model version. The batch size for the training dataset is set to 256 to ensure that a sufficient amount of data is processed in each training iteration. For the evaluation dataset, the batch size is set to 128 to balance the need for thorough evaluation with computational efficiency. Figure 1 illustrates the overall architecture and training process of the model:

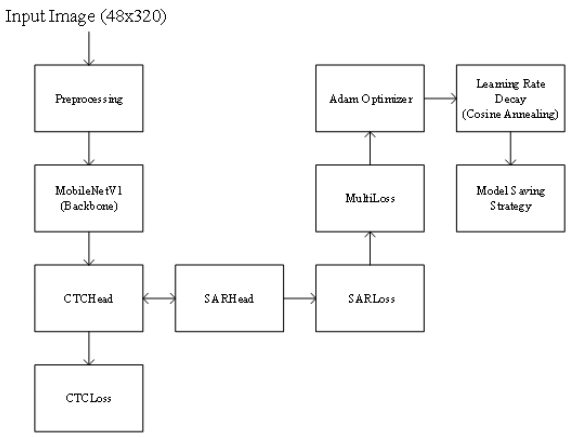

Figure 1. Overall Architecture and Training Process of the Model

DOI: 10.54254/2755-2721/81/20240967

4. Results

The experimental results show that the model performs well in the CAPTCHA recognition task. The best accuracy of 0.997 and the normalized edit distance of 0.999 indicate near-perfect character recognition. Despite the fact that float16 precision was not used in the training process, the model still achieved 2491 frames per second (fps), demonstrating its high processing efficiency. The training process showed rapid and stable convergence, with the best results observed at the 100th epoch.

A comparative analysis further highlights the model's excellence in CAPTCHA recognition tasks, with accuracy and normalized edit distance both approaching 1, ensuring high reliability and precision. The high fps makes it suitable for real-time applications requiring quick responses, such as automated form submissions, online security checks, and other verification systems. The model's robustness and flexibility are evident even in the absence of float16 accuracy, indicating that it can maintain high performance without compromising accuracy or speed.

Overall, the model performs exceptionally well in CAPTCHA recognition, achieving high accuracy and fast processing speed. These attributes make it suitable for practical applications, enhancing user experience and security in various online services. The successful integration of MobileNetV1 and the SVTR algorithm showcases the potential of deep learning in tackling complex CAPTCHA recognition tasks, offering a promising direction for future research and development in this field.

5. Discussion

5.1. Prospects of Deep Learning in CAPTCHA Recognition

Deep learning has broad application prospects in CAPTCHA recognition. Experimental results show that combining networks such as MobileNet for CAPTCHA recognition achieves high recognition speed while ensuring accuracy. For example, in this experiment, a processing speed of 2491.83 images per second was achieved. The accuracy of CAPTCHA recognition was also significantly improved, with the best accuracy reaching 0.9972 and the normalized edit distance close to 1, demonstrating the model's efficiency and stability. These results indicate that deep learning-based CAPTCHA recognition technology can not only be applied to current CAPTCHA recognition scenarios but also has the potential for broader applications in the future.

5.2. Challenges and Opportunities in CAPTCHA Technology Development

CAPTCHA technology faces several challenges in its development. One major issue is adversarial attacks, where hackers can use adversarial examples to deceive CAPTCHA recognition systems. The diversity and complexity of CAPTCHAs also present challenges, requiring more advanced recognition techniques. Additionally, performance and efficiency are challenges, necessitating improvements in recognition speed and efficiency while ensuring accuracy. However, CAPTCHA technology also holds many opportunities. The application of deep learning technology brings new possibilities for improving recognition accuracy and efficiency. The development of multimodal fusion technology can better utilize information from various CAPTCHA forms. Advances in intelligent hardware provide a broader application space for CAPTCHA technology. Interdisciplinary integration can promote cross-disciplinary cooperation in CAPTCHA technology, driving its development. While facing challenges, CAPTCHA technology also has opportunities for significant progress in the future.

5.3. Possible Research Directions and Innovations

The field of CAPTCHA recognition offers numerous avenues for experimental research, including the investigation of CAPTCHA recognition under adversarial attacks. Experiments can be conducted to ascertain the efficacy of adversarial sample generation and adversarial training methods in enhancing the security of CAPTCHA systems. Additionally, experiments can explore multimodal CAPTCHA recognition methods, improving recognition accuracy and robustness by integrating information from text, images, sound, etc. Moreover, experiments can attempt CAPTCHA generation technology based on deep learning, generate more complex and diverse CAPTCHAs using methods like Generative

DOI: 10.54254/2755-2721/81/20240967

Adversarial Networks (GANs), and verify their effect on enhancing the robustness of CAPTCHA recognition systems. In summary, experimental research can explore the development of CAPTCHA recognition technology in multiple aspects, providing important references for further research in the field.

6. Conclusion

In this paper, a deep learning-based CAPTCHA recognition system is proposed using MobileNetV1 to achieve high recognition speed and accuracy. The proposed system demonstrates that leveraging lightweight models and advanced algorithms can significantly enhance CAPTCHA recognition performance. This not only improves current CAPTCHA techniques but also contributes to developing more secure CAPTCHA generation mechanisms. The study of CAPTCHA recognition technology can elucidate its current limitations, optimize the existing CAPTCHA generation process, reinforce the security of the existing system, and propose novel concepts and methodologies for the prospective advancement of CAPTCHA technology. Future research may investigate the resistance of CAPTCHAs to attack, multimodal recognition, and the generation of CAPTCHAs based on deep learning, with the aim of further improving the capability of CAPTCHA technology.

References

[1] Lorenzi, D., Uzun, E., Vaidya, J. and Sural, S. (2018) Towards Designing Robust CAPTCHAs. Journal of Computer Security, 26(6): 731 - 760.

[2] Von Ahn, L., Blum, M., Hopper, N.J. and Langford, J. (2003) CAPTCHA: Using Hard AI Problems for Security. EUROCRYPT'03: Proceedings of the 22nd international conference on Theory and Applications of Cryptographic Techniques, 294-311.

[3] Chen S.Y. (2024) Complex Text CAPTCHA Recognition with Small Data Set. Electronic Design Engineering, 32(3): 54-58.

[4] Krizhevsky, A., Sutskever, I., Hinton, G.E. (2012) ImageNet Classification with Deep Convolutional Neural Networks. Advances in Neural Information Processing Systems, 25: 1097-1105.

[5] Gao, H.C., Wang, W., Qi, J., et al. (2013) The Robustness of Hollow CAPTCHAs. Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security,1075–1086.

[6] Kumar, M., Jindal, M.K., et al. (2021) A Systematic Survey on CAPTCHA Recognition: Types, Creation and Breaking Techniques. Archives of Computational Methods in Engineering, 1107-1136.

[7] Huang, K.Z., Hussain, A., Wang, Q.F. and Zhang, R. (2019) Deep Learning: Fundamentals, Theory and Applications. Cognitive Computation Trends (COCT), 2: 111-138.

[8] Ketkar, N. (2017) Convolutional Neural Networks. Springer International Publishing, 63-78.

[9] Huang, G., Liu, Z., et al. (2017) Densely Connected Convolutional Networks. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 4700-4708.

[10] He, K.M., Zhang, X.Y., Ren, S.Q. and Sun, J. (2016) Deep Residual Learning for Image Recognition. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 770-778.