Computer Modeling in Engineering & Sciences

Tech Science Press

Doi:10.32604/cmes.2025.059586

ARTICLE

A Dual-Layer Attention Based CAPTCHA Recognition Approach with Guided Visual Attention

Zaid Derea[1,2], Beiji Zou[1], Xiaoyan Kui[1,*], Alaa Thobhani[1] and Amr Abdussalam[3]

[1]School of Computer Science and Engineering, Central South University, Changsha, 410083, China
[2]College of Computer Science and Information Technology, Wasit University, Wasit, 52001, Iraq
[3]Electronic Engineering and Information Science Department, University of Science and Technology of China, Hefei, 230026, China

*Corresponding Author: Xiaoyan Kui. Email: xykui@csu.edu.cn

Received: 12 October 2024; Accepted: 10 January 2025; Published: 03 March 2025

ABSTRACT: Enhancing website security is crucial to combat malicious activities, and CAPTCHA (Completely Automated Public Turing tests to tell Computers and Humans Apart) has become a key method to distinguish humans from bots. While text-based CAPTCHAs are designed to challenge machines while remaining human-readable, recent advances in deep learning have enabled models to recognize them with remarkable efficiency. In this regard, we propose a novel two-layer visual attention framework for CAPTCHA recognition that builds on traditional attention mechanisms by incorporating Guided Visual Attention (GVA), which sharpens focus on relevant visual features. We have specifically adapted the well-established image captioning task to address this need. Our approach utilizes the first-level attention module as guidance to the second-level attention component, incorporating two LSTM (Long Short-Term Memory) layers to enhance CAPTCHA recognition. Our extensive evaluation across four diverse datasets—Weibo, BoC (Bank of China), Gregwar, and Captcha 0.3—shows the adaptability and efficacy of our method. Our approach demonstrated impressive performance, achieving an accuracy of 96.70% for BoC and 95.92% for Webo. These results underscore the effectiveness of our method in accurately recognizing and processing CAPTCHA datasets, showcasing its robustness, reliability, and ability to handle varied challenges in CAPTCHA recognition.

KEYWORDS: Text-based CAPTCHA image recognition; guided visual attention; web security; computer vision

1 Introduction

CAPTCHAs, or Completely Automated Public Turing tests to tell Computers and Humans Apart, play a crucial role in internet security by distinguishing between human users and automated bots [1–3]. Available in various formats such as text, image, audio, and video, CAPTCHAs are designed to strengthen online defenses against malicious activities [4–7]. Among these, text-based CAPTCHAs have become one of the most commonly used and effective methods for protecting online platforms due to their simplicity and ease of implementation across a wide range of applications [8,9]. However, the rapid advancements in technology, particularly in deep learning algorithms [10,11], have increasingly threatened the security of traditional CAPTCHA systems.

As machine learning techniques become increasingly sophisticated, CAPTCHA systems must continuously evolve to counter emerging threats. Adversarial attacks and advanced algorithms have driven the development of more resilient CAPTCHA designs, incorporating elements like distortion, rotation,

and context-based challenges to thwart automated recognition. Despite these advancements, the ongoing battle between CAPTCHA developers and attackers highlights the need for continuous innovation in online security measures. The rapid advancements in deep learning, known for its exceptional feature extraction capabilities, have significantly impacted domains like image restoration and object detection [12–15], making it a powerful tool for CAPTCHA recognition systems. However, this also poses challenges for text-based CAPTCHAs, as traditional methods struggle with feature extraction and are vulnerable to image noise. As a result, there is a growing trend towards deep learning-based CAPTCHA recognition, categorized into segmentation-based and segmentation-free approaches [16,17]. While segmentation-based methods involve dissecting characters before recognition and often face efficiency challenges, segmentation-free algorithms bypass this step, directly recognizing and classifying CAPTCHA characters with promising accuracy and efficiency.

Technologies like image captioning share similarities with CAPTCHA recognition, particularly in interpreting and processing visual data. However, image captioning methods have not been fully adapted for CAPTCHA recognition, which represents a missed opportunity to leverage these techniques for broader applications. One area that remains underexplored is the integration of advanced visual attention techniques, which could significantly enhance feature extraction and improve processing efficiency, ultimately boosting recognition accuracy. Furthermore, the potential of using multiple attention layers has not been adequately investigated in CAPTCHA recognition. Incorporating these layers could enable the model to better handle complex visual data, further improving accuracy. Additionally, traditional recognition methods often treat all parts of the image equally, limiting effective feature extraction. This issue is especially problematic in complex or noisy images, where focusing on the most relevant features is crucial for achieving successful recognition.

In this regard, we introduce a novel CAPTCHA recognition system called the Dual-Layer Attention-Based CAPTCHA Recognition Approach with Guided Visual Attention (DLACRM), as depicted in Fig. 1. Our model features a specialized recurrent neural network (RNN) architecture, adapted from the UpDown model [18], which is widely used in image captioning tasks. This adaptation is specifically designed to address the complexities of CAPTCHA recognition. To enhance the model's ability to detect subtle details in CAPTCHA images, we incorporate convolutional neural networks (CNNs) to extract both global and local features. Instead of relying solely on the conventional visual attention mechanism, we extend this approach by introducing a novel two-layer attention framework. The first layer employs traditional visual attention, while the second layer, termed Guided Visual Attention (GVA), further refines the focus on relevant visual features, thereby improving CAPTCHA recognition accuracy. Additionally, we integrate dual layers of Long Short-Term Memory (LSTM) networks in the decoder, which enhances the model's capability to accurately predict CAPTCHA characters. This strategic design results in a more refined and robust CAPTCHA recognition system, significantly improving the accuracy and reliability of decoding CAPTCHA images.

In particular, we propose a CAPTCHA recognition system that effectively addresses key challenges in traditional systems, such as character overlapping and distortion, which are increasingly vulnerable to advanced deep learning attacks. Current recognition models often struggle to capture the critical visual features necessary for accurate CAPTCHA decoding, as many rely on single-layer attention mechanisms that inadequately represent the depth of visual details, or they omit attention layers altogether, limiting feature extraction. To overcome these limitations, our system integrates a refined multi-layered attention mechanism, specifically the Guided Visual Attention (GVA) technique [19], along with a tailored UpDown model. This approach refines the output from the first attention layer by passing it to a second layer, allowing for deeper and more accurate feature representation. The guided attention dynamically adjusts the weights of the first layer to optimize the focus of the second, effectively mitigating issues like "attention drift" and enhancing the model's ability to handle complex visual data. This results in significantly improved

recognition accuracy across various complex CAPTCHA schemes, offering a robust and versatile solution to bolster online security against automated threats.

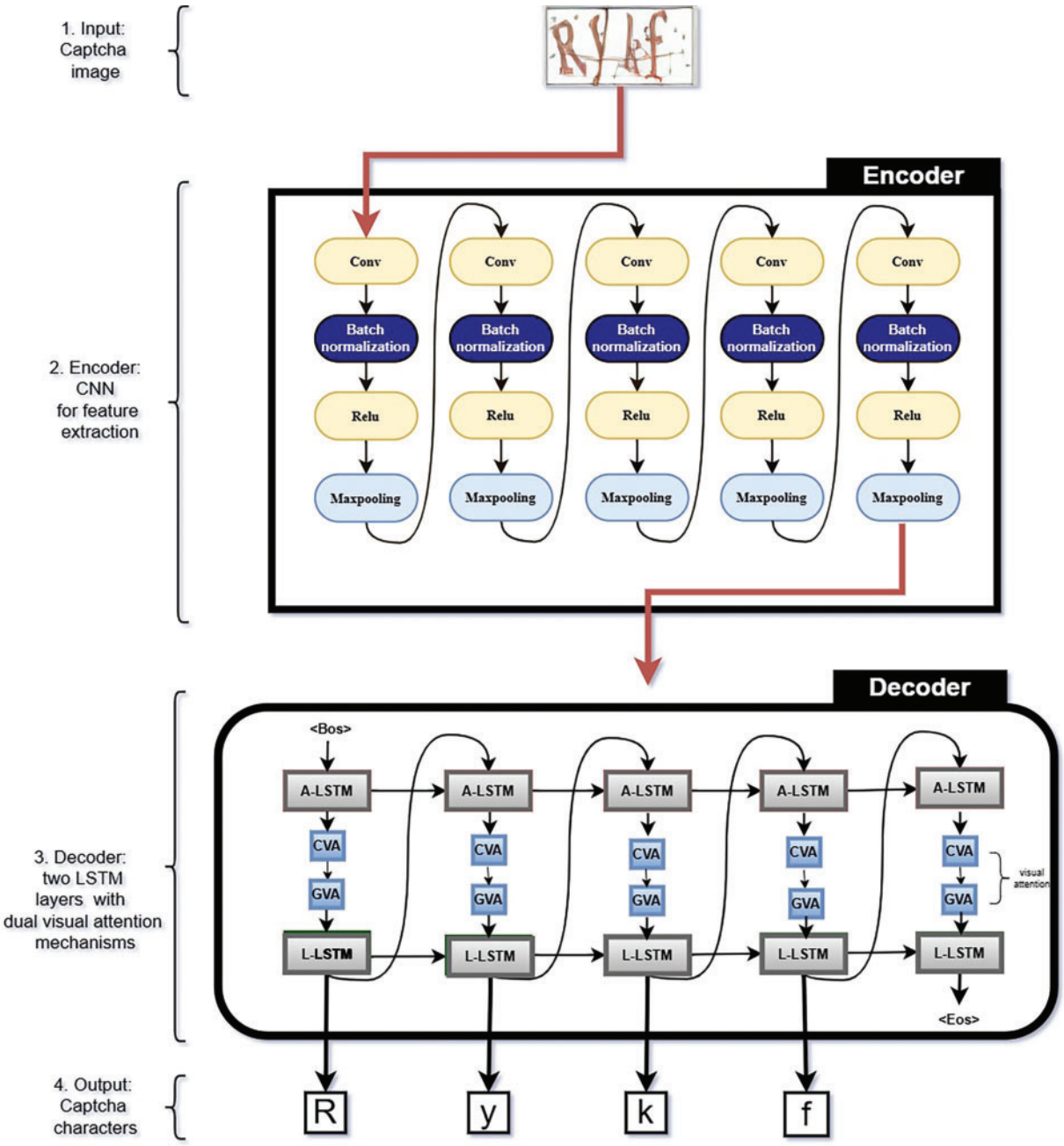

Figure 1: The diagram shows the overall process flow of the proposed CAPTCHA recognition framework

The evaluation conducted in this work encompasses datasets from diverse and prominent sources, including the Bank of China (BoC), Weibo, Gregwar, and Captcha 0.3, each representing a variety of CAPTCHA schemes. These datasets include manually collected BoC CAPTCHAs from the Bank of China's official website, Weibo CAPTCHAs from the Chinese social media platform Weibo, as well as CAPTCHAs

generated using the Gregwar CAPTCHA library and Captcha 0.3. Our thorough analysis reveals outstanding success rates in defeating these targeted CAPTCHAs without the need for segmentation, achieving 96.7% success on BoC CAPTCHAs and 95.92% on Weibo CAPTCHAs. This work introduces a groundbreaking deep-learning-driven CAPTCHA recognition system that excels in efficiency and simplicity, eliminating the need for segmentation. By leveraging image captioning techniques, the system achieves exceptional success in decoding text-based CAPTCHAs. The strength of our approach lies in its straightforwardness and versatility, making it a promising solution for enhancing online security. Our contributions include both theoretical advancements in CAPTCHA recognition and practical applications aimed at strengthening internet security.

This work presents several key contributions:

- We explore the use of image captioning techniques for CAPTCHA recognition and employing more than one visual attention layer.
- We utilize a two-layer visual attention mechanism. The first layer applies conventional visual attention, which serves as guidance for the second module, termed Guided Visual Attention (GVA), aimed at enhancing CAPTCHA recognition.
- We propose a novel model that encompasses a CNN-based encoder, a Guided Visual Attention module (GVA), a Conventional Visual Attention module (CVA), and two LSTM layers to enhance the CAPTCHA recognition process.
- Our study involves comprehensive experiments across four different dataset schemes, including two real-world datasets, Weibo and BoC, which were carefully collected and manually labeled. Furthermore, our proposed approach shows performance on par with recent state-of-the-art methods.

The structure of this paper is as follows: Section 2 reviews various CAPTCHA recognition methods and algorithms. Section 3 outlines the fundamental concept of the proposed CAPTCHA recognition algorithm, along with the architecture and parameters of the recognition CNN. In Section 4, the paper describes the structure of the datasets used, evaluates the accuracy of the proposed CAPTCHA recognition model, compares the results, and discusses the proposed algorithm. Finally, Section 5 presents the conclusion of the study.

2 Related Work

CAPTCHA (Completely Automated Public Turing test to tell Computers and Humans Apart) is a security mechanism designed to distinguish between human users and automated bots. It plays a critical role in preventing spam, data scraping, and unauthorized access to websites and online services [20]. CAPTCHA challenges come in various forms, such as text-based, image-based, and audio-based, with each type presenting a task that is easy for humans but difficult for machines to solve [21–23]. Text-based CAPTCHA, the most traditional type, typically requires users to identify and input distorted or obfuscated characters displayed in an image [24–26]. This method leverages the human brain's superior ability to recognize patterns despite visual noise, which bots struggle to replicate [27,28]. Despite advancements in machine learning and optical character recognition (OCR) systems, text-based CAPTCHAs remain a widely used tool for online security due to their simplicity and effectiveness in countering automated threats [29,30].

Text-based CAPTCHA systems have undergone significant evolution due to advancements in machine learning and artificial intelligence. Early approaches focused on distorting text with noise, geometric transformations, and background clutter, which posed a challenge for Optical Character Recognition (OCR) systems. However, deep learning models, especially Convolutional Neural Networks (CNNs), have dramatically improved the accuracy of CAPTCHA solvers, with success rates often exceeding 90% for even

complex challenges. Research demonstrated the vulnerability of text-based CAPTCHA systems to CNN-based attacks, leading to further exploration of hybrid models combining CNNs with Recurrent Neural Networks (RNNs) or Long Short-Term Memory (LSTM) networks to improve recognition of overlapping and distorted characters [31]. More recent works have explored the use of Convolutional Recurrent Neural Networks (CRNNs) to tackle complex text-based CAPTCHAs [21]. Adversarial machine learning techniques, such as those proposed by Shi et al. [20], are now being used to enhance the security of CAPTCHAs by generating adversarial examples that are challenging for machines but easily interpretable by humans [20]. Despite these advancements, the growing power of machine learning necessitates continued innovation in CAPTCHA design, particularly through hybrid and adversarial approaches to maintain security and usability [10,32,33].

Thobhani et al. introduced an innovative method using Convolutional Neural Networks (CNNs) with binary images, which demonstrated exceptional accuracy and significantly reduced system size [34]. However, despite its strengths, the approach faces challenges in CAPTCHA recognition, particularly due to the extensive need for annotated training data. In contrast, Derea et al. [35] proposed the CRNGS algorithm, which integrates deep learning with character grouping techniques to streamline CAPTCHA recognition without the necessity of image segmentation. This method leverages adaptable softmax layers, allowing for performance optimization across different CAPTCHA formats. Similarly, Khatavkar et al. [36] developed a segmentation-free Optical Character Recognition (OCR) model, utilizing the Connectionist Temporal Classification (CTC) loss function to efficiently classify text-based CAPTCHAs without the need for explicit segmentation of characters. On the other hand, Chang et al. [37] provided a comprehensive examination of security vulnerabilities in slider-based behavior-verification CAPTCHAs, a relatively underexplored area. They introduced a universal framework for accurately detecting target trajectories and simulating user behaviors, thus enhancing the robustness of these CAPTCHA systems [34–37]. Reference [38] presents a comparative analysis of text-based and image-based CAPTCHA systems, emphasizing the evaluation of both usability (solving time) and efficiency (response time) as key performance metrics.

Anderson et al. [18] proposed a novel approach that integrates both bottom-up and top-down attention mechanisms to improve visual question-answering and image captioning systems. This method, inspired by human visual perception, aims to replicate the way humans interpret images and respond to related questions. Numerous studies have built upon [18] in their image captioning research, including [39], introduces a dynamic approach that adjusts semantic attributes based on contextual relevance, enhancing the alignment between visual and textual information. The combination of the Attribute Detection Component (ADC) and the Attribute Prediction and Visual Weighting (APVW) module optimizes using attributes for generating more accurate captions. Reference [40] presents a two-step method for improving image captioning. Initially, the Visual Feature Detector (VFD) identifies key visual elements. Subsequently, the Visual Feature Visual Attention (VFVA) module concentrates on these features to refine the context, resulting in more precise captions. Reference [19] introduces the Guided Visual Attention (GVA) technique for generating image captions, enhancing the quality of the captions by refining how the attentional focus is distributed.

The existing literature shows notable gaps in adapting image captioning techniques for CAPTCHA recognition despite the similarities between the two technologies in processing visual data. Specifically, integrating advanced visual attention mechanisms, such as multiple attention layers, still needs to be explored in CAPTCHA recognition. These methods could significantly improve feature extraction, processing efficiency, and recognition accuracy, particularly in complex or noisy images. Moreover, traditional recognition approaches often fail to prioritize relevant features, limiting their effectiveness in challenging CAPTCHA scenarios.

3 Methodology

In our comprehensive study of CAPTCHA recognition, we propose a novel two-layer attention framework. This framework is designed to enhance CAPTCHA recognition by effectively focusing on relevant visual features. Specifically, the first layer of our attention framework employs conventional visual attention techniques. This initial layer guides the second layer, which we refer to as Guided Visual Attention (GVA). The GVA layer refines the attention mechanism to improve the precision of feature extraction and recognition. The proposed network utilizes a two-layer LSTM architecture to enhance the recognition process by exploiting temporal dependencies. A schematic overview of our proposed model is illustrated in Fig. 1.

3.1 Visual Features of Input Image

In our CAPTCHA recognition system, the first step involves extracting visual features from the input image, which are then used by the language model for further processing. The initial phase of CAPTCHA recognition focuses on deriving visual representations of the image. This is achieved using a Convolutional Neural Network (CNN) in the encoder component of our model to generate essential image features, denoted as V, for CAPTCHA identification. Specifically, these features are extracted from the output of the final max-pooling layer in the CNN, which consists of five convolutional layers and five max-pooling layers. The CNN uses the ReLU activation function and applies batch normalization. The structure of this CNN is depicted in Table 1. The CNN produces N visual feature vectors, forming the visual matrix $V \in \mathbb{R}^{N\times h}$. Each visual feature vector is represented as $v_i \in \mathbb{R}^h$ for $i \in \{1, 2, ..., N\}$. The visual matrix V obtained from the input image I via the CNN network is described by:

$$V = \{v_1, v_2, \ldots, v_N\} \tag{1}$$

$$\bar{v} = \frac{1}{N}\sum_{i=1}^{N} v_i \tag{2}$$

where $V \in R^{N\times h}$, $v_i \in R^h$ for $i \in \{1, 2, \ldots, N\}$, and $\bar{v} \in R^h$.

Table 1: The architecture of the CNN used in our model

Layer type	Filter size	Number of filters	Activation function
Convolutional	3 × 3	64	ReLU
Batch normalization	–	64	–
Maxpooling	2 × 2	–	–
Convolutional	3 × 3	128	ReLU
Batch normalization	–	128	–
Maxpooling	2 × 2	–	–
Convolutional	3 × 3	256	ReLU
Batch normalization	–	256	–
Maxpooling	2 × 2	–	–
Convolutional	3 × 3	512	ReLU
Batch normalization	–	512	–
Maxpooling	2 × 2	–	–
Convolutional	3 × 3	1024	ReLU

(Continued)

Table 1 (continued)

Layer type	Filter size	Number of filters	Activation function
Batch normalization	–	1024	–
Maxpooling	2 × 2	–	–

3.2 Conventional Visual Attention: CVA

In our DLACRM, we have adopted the conventional visual attention mechanism [18] as the first layer attention component. The output of the first layer attention module, known as the context visual vector $\hat{v}_t$, is utilized as guidance for the second layer attention component (introduced in Section 3.3), as shown in Fig. 2. The conventional visual attention mechanism is described by the following formulas:

$$\beta_t^i = \tanh\left(h_t^a \cdot W_a + v_i \cdot W_b\right) \cdot W_c \tag{3}$$

$$\alpha_t = \text{softmax}\left(\beta_t\right) \tag{4}$$

$$\hat{v}_t = \sum_{i=1}^{N} \left(\alpha_t^i \odot v_i\right) \tag{5}$$

where $\beta_t \in R^N$, and $\alpha_t \in R^N$. The hidden state of the attention LSTM $h_t^a \in R^g$, and the context visual vector of the first attention layer $\hat{v}_t \in R^h$. $W_a \in R^{e\times g}$, $W_b \in R^{e\times h}$, and $W_c \in R^e$ are trainable weights.

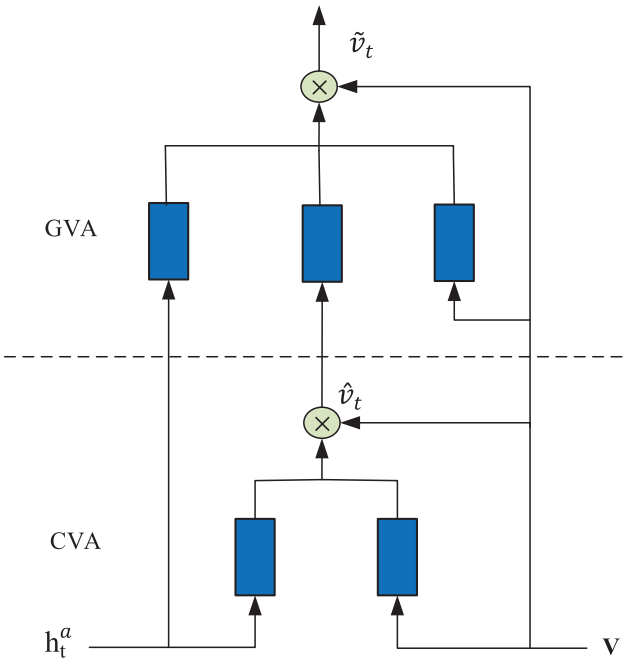

Figure 2: The structure of the two-layer visual attention module

3.3 Guided Visual Attention: GVA

In the task of CAPTCHA image recognition, the objective is to generate a sequence of characters for the CAPTCHA $C = (y_1, y_2, \ldots, y_T)$, which accurately reflects the content of the image I. Here, y_i represents a character in the CAPTCHA sequence C, and T is the total number of characters in the CAPTCHA. To achieve this, we utilize the Guided Visual Attention (GVA) mechanism, designed to effectively capture and process the visual features of the CAPTCHA image [19]. GVA operates as a second-level attention mechanism, assigning importance scores to the candidate feature vectors extracted from the image. These scores are then normalized using the softmax function to derive attention weights. The resulting attention-weighted feature vector is fed into the language LSTM module for character prediction. This process allows the model to focus on the most relevant parts of the image, ensuring accurate character generation. The internal structure of the GVA module is depicted in Fig. 3. As shown in Fig. 4, our approach leverages a two-layer LSTM architecture: the attention LSTM for managing the visual context and the language LSTM for generating the final sequence of characters, which is defined as:

$$\delta_t^i = \tanh\left(W_e \cdot h_t^a + W_f \cdot v_i + W_g \cdot \hat{v}_t\right) \cdot W_d \tag{6}$$

$$\gamma_t = softmax\left(\delta_t\right) \tag{7}$$

$$\tilde{v}_t = \sum_{i=1}^{N} \left(\gamma_t^i \odot v_i\right) \tag{8}$$

where $\delta_t \in R^N$, and $\gamma_t \in R^N$. The context visual vector of the GVA attention module is $\tilde{v}_t \in R^h$. $W_d \in R^e$, $W_e \in R^{e \times g}$, $W_f \in R^{e \times h}$, and $W_g \in R^{e \times h}$ are learnable parameters.

3.4 Language Model

The decoder part of our CAPTCHA recognition model encompasses mainly two LSTM layers along with the two layers of visual attention modules. The hidden state of the attention LSTM $h_t^a \in R^g$ is given by:

$$h_t^a = LSTM_a\left(h_{t-1}^a, \left[h_{t-1}^l, E \cdot y_{t-1}, \bar{v}\right]\right) \tag{9}$$

where $h_{t-1}^l \in R^g$ represents the hidden state of the language LSTM in the previous time step, $y_{t-1} \in R^c$ is the generated character in the previous time step, and $E \in R^{c \times m}$ signifies the character embedding matrix. The language LSTM network receives input from both the output of the attention LSTM h_t^a and the context vector generated by the Guided Visual Attention (GVA) mechanism $\tilde{v}_t$. Consequently, the output of the language LSTM is defined as:

$$h_t^l = LSTM_l\left(h_{t-1}^l, \left[\tilde{v}_t, h_t^a\right]\right) \tag{10}$$

where $h_t^l \in R^g$ is the hidden state of the language LSTM. The output of the language LSTM is utilized through the softmax function to predict the next character, which is defined as:

$$p_t = \text{softmax}\left(W_p \cdot h_t^l\right) \tag{11}$$

where $p_t \in R^c$ is the probability distribution along the whole characters, and $W_p \in R^{c \times g}$ is a trainable parameter matrix.

Concisely, the extracted features V and h_t^a are first sent to the visual attention module, which generates the output feature $\hat{v}_t$. Then, $\hat{v}_t$,V, and h_t^a are passed through the proposed Guided Visual Attention component to produce the visual vector representation $\tilde{v}_t$. Next, $\tilde{v}_t$ and the hidden state h_t^a are fed into the language LSTM module. The output of this module goes through a fully connected layer, followed by

a softmax layer, which yields a probability distribution with the highest probability corresponding to the predicted character y_t. At each time step, the predicted characters are collected to form the final CAPTCHA.

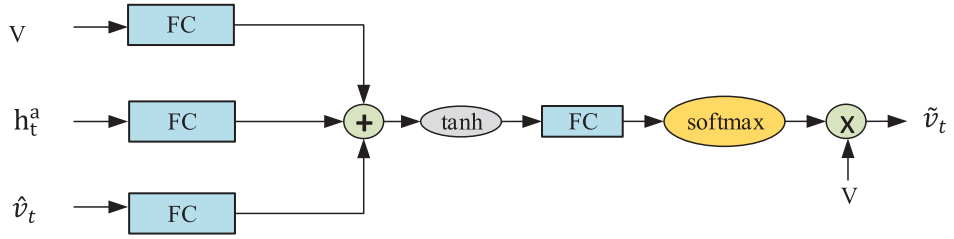

Figure 3: The diagram depicts the internal structure of the Guided Visual Attenion (GVA)

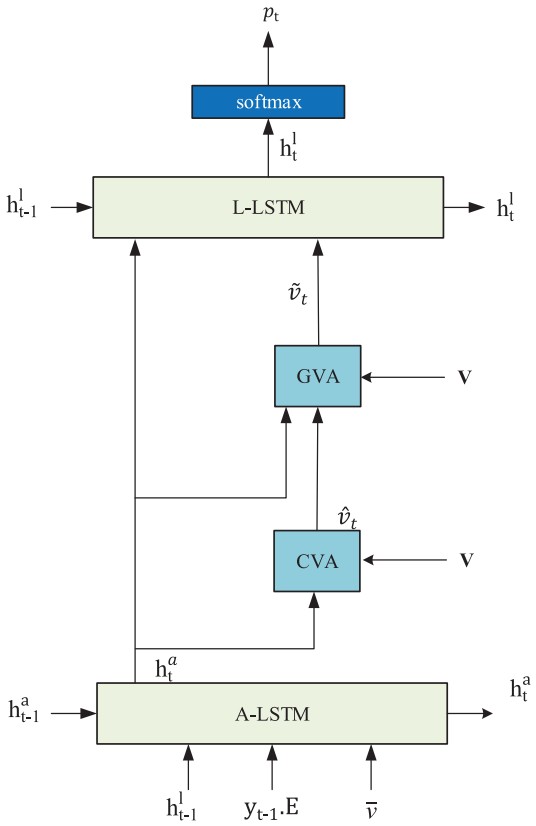

Figure 4: The diagram displays the internal structure of the language model, where A-LSTM represents the attention LSTM and L-LSTM denotes the language LSTM

3.5 Loss Functions

We utilize cross-entropy (XE) for training our model, employing the standard cross-entropy loss $Loss_{XE}$, which is defined as:

$$\text{loss}_{XE} = \frac{1}{T}\sum_{t=1}^{T} -\log\left(p_t\left(y_t \mid y_{1:t-1}, V\right)\right) \tag{12}$$

4 Experiments and Results

In this section, we furnish detailed insights into the datasets utilized for training, validating, and evaluating the DLACRM model. After delving into the datasets, we provide an exhaustive description of the CNN architecture and attention mechanism employed within the DLACRM model, alongside a comprehensive overview of the training parameters applied. Subsequent to setting up the model, we meticulously evaluate its accuracy using various metrics. Furthermore, we conduct an extensive comparative analysis, juxtaposing the performance of the DLACRM algorithm with that of other existing CAPTCHA recognition systems. This comparative scrutiny not only underscores the strengths of the DLACRM model but also pinpoints areas for potential enhancement, thus offering valuable insights into its overall effectiveness and limitations.

4.1 Used Datasets

Due to the limited availability of CAPTCHA datasets, sourcing CAPTCHA images is critical for recognition research. We collect images through two methods: extracting from live online platforms and using CAPTCHA generation software. Our study uses four datasets: Bank of China (BoC) (https://ebsnew.boc.cn/boc15/login.html/) (accessed on 09 January 2025), Weibo (https://www.weibo.com/) (accessed on 09 January 2025), Captcha 0.3, and Gregwar. Fig. 5 shows examples from these datasets, highlighting their diversity.

Figure 5: Illustrative examples of CAPTCHA schemes employed within the DLACRM model

4.1.1 Bank of China CAPTCHA Dataset

To address the growing threat of automated attacks, the Bank of China, which operates over 10,000 branches worldwide, has implemented an advanced CAPTCHA system. This system features overlapping characters, distortion, warping, and rotation, all designed to challenge bots. Each CAPTCHA consists of four characters from a set of uppercase English letters and numbers, with characters like G, C, Q, I, O, L, S, 0, 1,

and 5 excluded to minimize risk. Our dataset, sourced from the Bank's CAPTCHA system, includes 70,000 varied CAPTCHA images. By sharing this dataset, the Bank of China supports research and the development of improved CAPTCHA recognition technologies, underscoring its commitment to enhancing cybersecurity and protecting financial data.

4.1.2 Weibo CAPTCHA Dataset

As of 2022, Weibo, one of China's top social media platforms, has 586 million monthly active users and is recognized for its commitment to online security. Weibo employs a CAPTCHA system with features like character overlap, distortion, warping, and rotation to protect user accounts. The CAPTCHAs consist of four characters, excluding specific letters and digits (e.g., I, D, G, U, Q, 0, 1, and 5) to strengthen security against bots. Our dataset of 70,000 Weibo CAPTCHA images is carefully labeled for analysis, providing insights into the system's effectiveness and supporting the development of advanced security algorithms. This collection underscores Weibo's proactive approach to online security.

4.1.3 Captcha 0.3 CAPTCHA Dataset

An open-source tool, Captcha 0.3 allows users to easily generate custom CAPTCHAs. For our dataset, we opted for a four-character format, drawing from a mix of numeric digits (0–9) and both uppercase and lowercase English letters (A–Z, a–z), resulting in a diverse range of combinations. Our collection consists of 70,000 unique CAPTCHA images, each generated with complete randomness to ensure no duplicates. To improve security, we incorporated intersecting lines over the characters and added noisy dots in the background. The CAPTCHAs were created using the "liberbaskerville-regular" font. Fig. 5 presents sample images from our dataset, showcasing the variety and security features included in the CAPTCHA design.

4.1.4 Gregwar CAPTCHA Dataset

Known for its strong CAPTCHA generation, the Gregwar PHP library offers effective defense against automated bot attacks. By incorporating security elements like intricate noise lines, stylish backgrounds, and rotational effects, it generates CAPTCHAs that can withstand even persistent bot threats. Each CAPTCHA is made up of four randomly selected characters from three categories: numeric digits, uppercase, and lowercase English letters, presenting a significant challenge for bots. Our dataset, which has been meticulously compiled, features 70,000 unique Gregwar CAPTCHA images. The random character selection ensures that no duplicates or repetitions occur, enhancing the dataset's diversity and making it a valuable resource for testing CAPTCHA recognition systems and evaluating security measures.

4.1.5 Preprocessing Steps

Each CAPTCHA dataset is systematically divided into three distinct subsets: a comprehensive training set consisting of 50,000 varied CAPTCHA images, a carefully curated testing set with 10,000 selected CAPTCHA images, and a thorough validation set comprising 10,000 meticulously chosen CAPTCHA images. Each CAPTCHA image across these datasets has a filename that includes a label representing a unique four-character string extracted from the CAPTCHA itself. During the initial preprocessing phase, all CAPTCHA images are converted to grayscale and resized to uniform dimensions of 64×256 to ensure consistency throughout the dataset. It is important to note that the selection process for the images in the training, validation, and testing sets is carried out randomly to eliminate potential biases and maintain the dataset's integrity for accurate evaluation and training purposes.

4.2 Experimental Settings

We employ a CNN architecture (as shown in Table 1) to extract features from images, producing object representations with dimensions of 16×1024. For the CAPTCHA recognition task, each input image is represented by a visual feature vector of size $h = 1024$. The LSTM networks are designed with a hidden state size of $g = 1000$ to capture complex linguistic structures during CAPTCHA generation, mapping each character to a vector of length $f = 1000$. The model processes $N = 16$ visual feature vectors per image, with an internal attention mechanism of size $e = 512$ to focus on important regions of the image. The number of characters c varies across CAPTCHA schemes: 28 for Weibo, 26 for BoC, and 62 for both Gregwar and Captcha 0.3. For model training, we use the Adam optimizer with an initial learning rate of 0.0005, which decays by 0.8 every 5 epochs over a total of 120 epochs. The batch size is set to 50, and scheduled sampling is increased by 5% every 5 epochs, capping at 25%. Gradient clipping is applied with a maximum value of 0.1, and dropout is used at a rate of 0.5. Testing employs a beam size of 3 with a beam search strategy, and the model is implemented in the PyTorch framework.

4.3 Model Accuracy

Table 2 displays the comprehensive CAPTCHA accuracies of the DLACRM model across different schemes, including Weibo, BoC, Gregwar, and Captcha 0.3 CAPTCHA. Notably, overall accuracy refers to the percentage of correctly identified CAPTCHAs across the entire dataset, where each CAPTCHA is considered correct only if all characters are correctly predicted. In contrast, total accuracy measures the percentage of correctly predicted individual characters across all CAPTCHAs, regardless of whether the entire CAPTCHA string is correctly identified. While overall accuracy evaluates the model's performance at the CAPTCHA level, total accuracy provides insight into the model's precision at the character level, highlighting its ability to correctly recognize individual characters even if the full CAPTCHA is not identified accurately. Before delving into the exceptional results achieved with the BoC CAPTCHA, it's important to highlight the DLACRM model's performance with the Weibo CAPTCHA scheme. The model attained an impressive accuracy of 95.92%, successfully recognizing 9,592 out of 10,000 images. This demonstrates the model's robustness in handling different CAPTCHA formats, establishing a solid foundation for its versatility.

Table 2: The four CAPTCHA schemes—Gregwar, Captcha 0.3, Weibo, and BoC—used to assess both individual character accuracy and overall CAPTCHA accuracy for DLACRM

	Gregwar scheme	**Captcha 0.3 scheme**	**Webo scheme**	**BoC scheme**
1st character accuracy	79.38% (7938/10,000)	98.89% (9889/10,000)	98.74% (9874/10,000)	98.97% (9897/10,000)
2st character accuracy	71.95% (7195/10,000)	97.44% (9744/10,000)	98.59% (9859/10,000)	99.86% (9986/10,000)
3st character accuracy	71.42% (7142/10,000)	96.54% (9654/10,000)	98.68% (9868/10,000)	99.84% (9984/10,000)
4st character accuracy	82.24% (8224/10,000)	98.28% (9828/10,000)	98.89% (9889/10,000)	99.20% (9920/10,000)
Total character accuracy	76.24% (30,499/40,000)	97.78% (39,115/40,000)	98.72% (39,490/40,000)	99.06% (39,627/40,000)
Overall CAPTCHA Accuracy	40.57% (4057/10,000)	93.76% (9376/10,000)	95.92% (9592/10,000)	96.70% (9670/10,000)

Moving on to the BoC CAPTCHA scheme, the DLACRM model truly excelled, achieving a remarkable accuracy rate of 96.70%. This translated to correctly identifying 9670 out of 10,000 images. The model's success here underscores its ability to manage the intricacies of the BoC CAPTCHA, solidifying its efficacy in more complex scenarios. Similarly, in the Captcha 0.3 scheme, the DLACRM model continued its high performance, achieving a recognition accuracy of 93.76%. Out of 10,000 images, it accurately identified 9376, demonstrating consistent reliability across varying CAPTCHA designs. However, the Gregwar CAPTCHA scheme posed a significant challenge for the model. Here, the DLACRM model's accuracy dropped to 40.57%, with correct recognition of 4057 out of 10,000 images. Despite this lower accuracy, the model's capacity to decipher a substantial number of Gregwar CAPTCHAs suggests room for improvement and adaptation in tackling more complex CAPTCHA types.

Looking at the overall character accuracy across different CAPTCHA schemes, the DLACRM model showcased robust performance. In the BoC CAPTCHA scheme, it achieved an outstanding character accuracy of 99.06%, accurately identifying 39,627 out of 40,000 characters. This result highlights the model's exceptional precision in deciphering BoC CAPTCHAs. In the Weibo CAPTCHA scheme, the model maintained a high accuracy rate of 98.72%, with 39,490 characters recognized correctly out of 40,000. The Captcha 0.3 scheme also displayed strong accuracy, achieving a rate of 97.78%, with 39,115 characters accurately identified. These results further validate the model's effectiveness across diverse CAPTCHA formats.

Even in the more challenging Gregwar CAPTCHA scheme, the model achieved a commendable character accuracy of 76.24%, correctly identifying 30,499 characters out of 40,000. Despite the inherent complexity of Gregwar CAPTCHAs, this performance showcases the model's adaptability and potential for refinement.

When examining individual character accuracies, the DLACRM model exhibited remarkable performance across all CAPTCHA schemes, demonstrating its robustness in character recognition. In the BoC CAPTCHA, the model achieved character-specific accuracies of 98.97%, 99.86%, 99.84%, and 99.20% for the first through fourth characters, respectively. Similarly, in the Weibo CAPTCHA scheme, accuracy rates were consistently high, with 98.74%, 98.59%, 98.68%, and 98.89% for the respective characters. The Captcha 0.3 scheme showed stable results as well, with individual accuracies of 98.89%, 97.44%, 96.54%, and 98.28%. Despite the difficulties posed by the Gregwar CAPTCHA scheme, the model still achieved notable character accuracy rates: 79.38%, 71.95%, 71.42%, and 82.24% for the first through fourth characters. This underscores the DLACRM model's resilience, even in the face of more complex challenges.

The training performance of the DLACRM model on the BoC dataset is highlighted through total and overall accuracy metrics, which are presented in Fig. 6. Over the 120 training epochs, the model shows significant improvement in performance. Initially, at epoch 0, both total and overall accuracy were at 0%, indicating no initial accuracy. By epoch 3, the total accuracy reached 48%, and overall accuracy was at 18%, demonstrating a rapid early learning phase. As training progressed, the total accuracy continued to improve, reaching 92% by epoch 6 and stabilizing at an impressive 97% by epoch 9. From epoch 12 onward, the total accuracy maintained a high level of 98%, indicating the model's strong and consistent learning capabilities. The overall accuracy followed a more gradual increase, reaching 83% by epoch 6, 92% by epoch 9, and stabilizing at 95% by epoch 12. This gradual but consistent improvement underscores the model's robust ability to generalize across various tasks, maintaining high accuracy through the training phases. The model's rapid convergence and sustained performance highlight its effectiveness in character recognition tasks and broader evaluation criteria.

Fig. 7 provides the training performance of the DLACRM model on the BoC dataset, with a specific focus on the accuracy for Characters 1, 2, 3, and 4 during the 120 epochs. Character 1 demonstrated rapid learning during training, starting at 0% accuracy at epoch 0 and reaching 58% by epoch 3. It surged to 95%

by epoch 6 and stabilized at 98% from epoch 9 onwards, showing robust performance. Character 2 had a slightly slower start, beginning at 0% and reaching 42% by epoch 3, but improved quickly to 92% by epoch 6 and 97% by epoch 9, maintaining 98% accuracy thereafter.

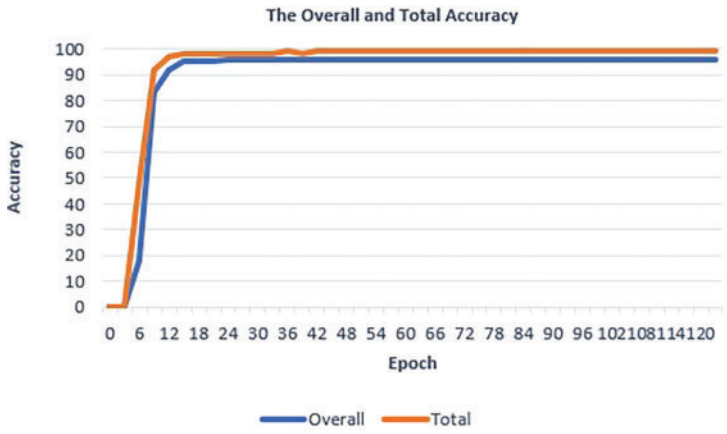

Figure 6: The overall and total character recognition accuracy of DLACRM on the BoC dataset during the training phase

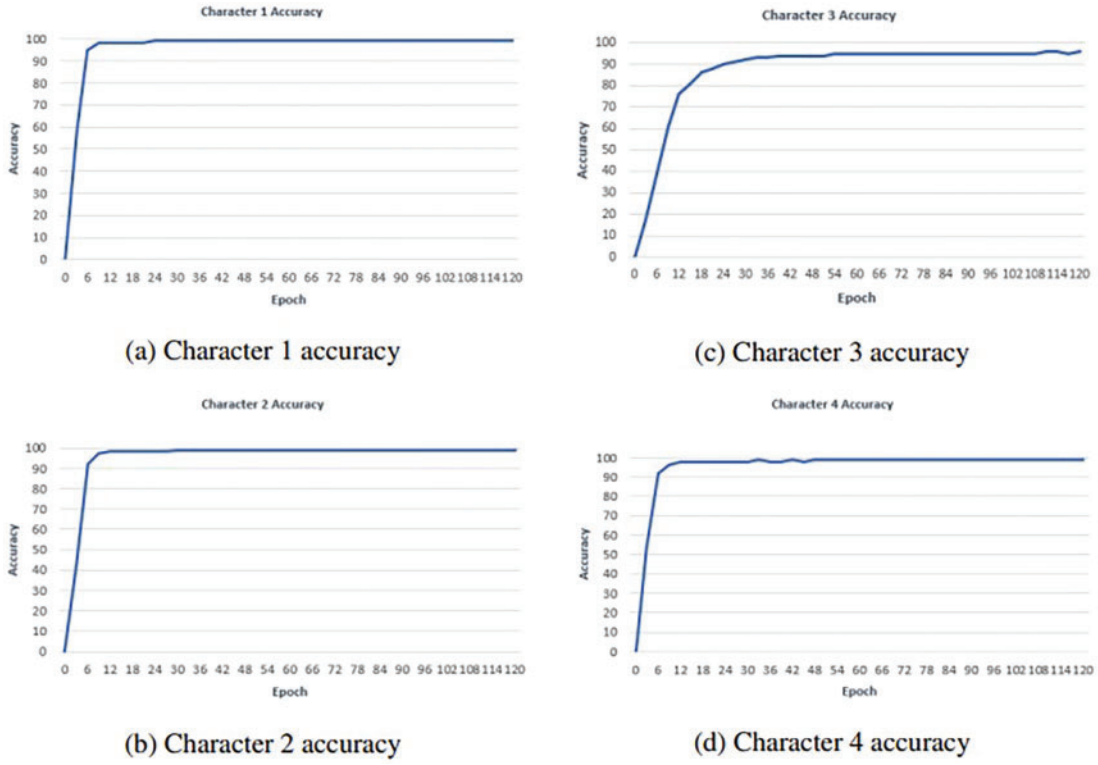

Figure 7: Character accuracy for Characters 1 (a), 2 (b), 3 (c) and 4 (d) using the DLACRM model on the BoC dataset during the training phase

Character 3 started at 0%, improved to 40% by epoch 3, and rose rapidly to 89% by epoch 6 and 95% by epoch 9, stabilizing at 97% from epoch 12 onwards. Character 4 followed a similar trend to Character 1, starting at 0%, reaching 54% by epoch 3, 92% by epoch 6, and 96% by epoch 9, with a consistent accuracy of 98% from epoch 12. All characters showed significant improvements in the early epochs, stabilizing at high accuracy levels (97-98%) by epoch 12, highlighting the model's robust learning and consistent performance across diverse character recognition tasks.

The training performance of DLACRM model on the Weibo dataset, focusing on total and overall accuracy metrics, demonstrates a clear progression of learning over the 120 epochs, as presented in Fig. 8. Initially, at epoch 0, both total and overall accuracies were at 0%, reflecting no initial performance. By epoch 3, the total accuracy had increased to 26%, while overall accuracy remained low at 1%, indicating the early stages of the model's learning process. As training progressed, the total accuracy showed substantial improvement, reaching 54% by epoch 6 and stabilizing at 84% by epoch 9. From epoch 12 onward, the total accuracy consistently maintained a high level of 93%, highlighting the model's effective learning and adaptation capabilities. The overall accuracy followed a similar trend but with a more gradual improvement, starting at 21% by epoch 6, reaching 65% by epoch 9, and stabilizing at 83% from epoch 12 onwards. These metrics underscore the model's growing proficiency and robust performance across the training period, with rapid convergence and sustained high performance from the mid-training stages onwards.

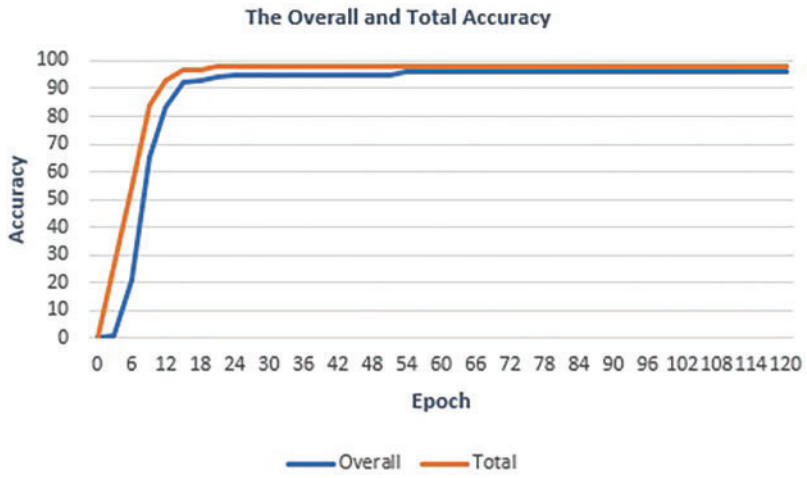

Figure 8: The overall character recognition accuracy of DLACRM on the Weibo dataset during the training phase

Fig. 9 The provides the training performance of the DLACRM model on the Weibo dataset, with a specific focus on the accuracy for Characters 1, 2, 3, and 4 during the 120 epochs. Character 1 demonstrated significant learning progress during training, starting at 0% accuracy at epoch 0 and improving to 32% by epoch 3. By epoch 6, Character 1's accuracy had increased sharply to 74%, and by epoch 9, it reached 92%, eventually stabilizing at 96% from epoch 12 onwards, indicating a strong and rapid learning curve. Character 2 began with a lower initial accuracy of 0% and reached 22% by epoch 3. Despite this slower start, Character 2 showed a steady improvement, achieving 54% by epoch 6 and 83% by epoch 9, before stabilizing at 93% from epoch 12, demonstrating the model's effective adaptation even with initial variability. Character 3 faced a more challenging start, beginning at 0% and improving to 20% by epoch 3. However, the accuracy quickly increased to 41% by epoch 6 and 77% by epoch 9, stabilizing at 90% from epoch 12 onwards, reflecting consistent growth and adaptation. Character 4 showed a pattern similar to Character 1, starting at 0% and reaching 30% by epoch 3. By epoch 6, Character 4's accuracy had improved to 47%, and by epoch

9, it reached 82%, stabilizing at 93% from epoch 12 onwards, highlighting the model's robust performance across all characters. Overall, each character experienced substantial improvements in the early epochs, with high stabilization levels (90%–96%) by epoch 12, underscoring the model's strong learning capabilities and consistent performance in character recognition tasks across the dataset.

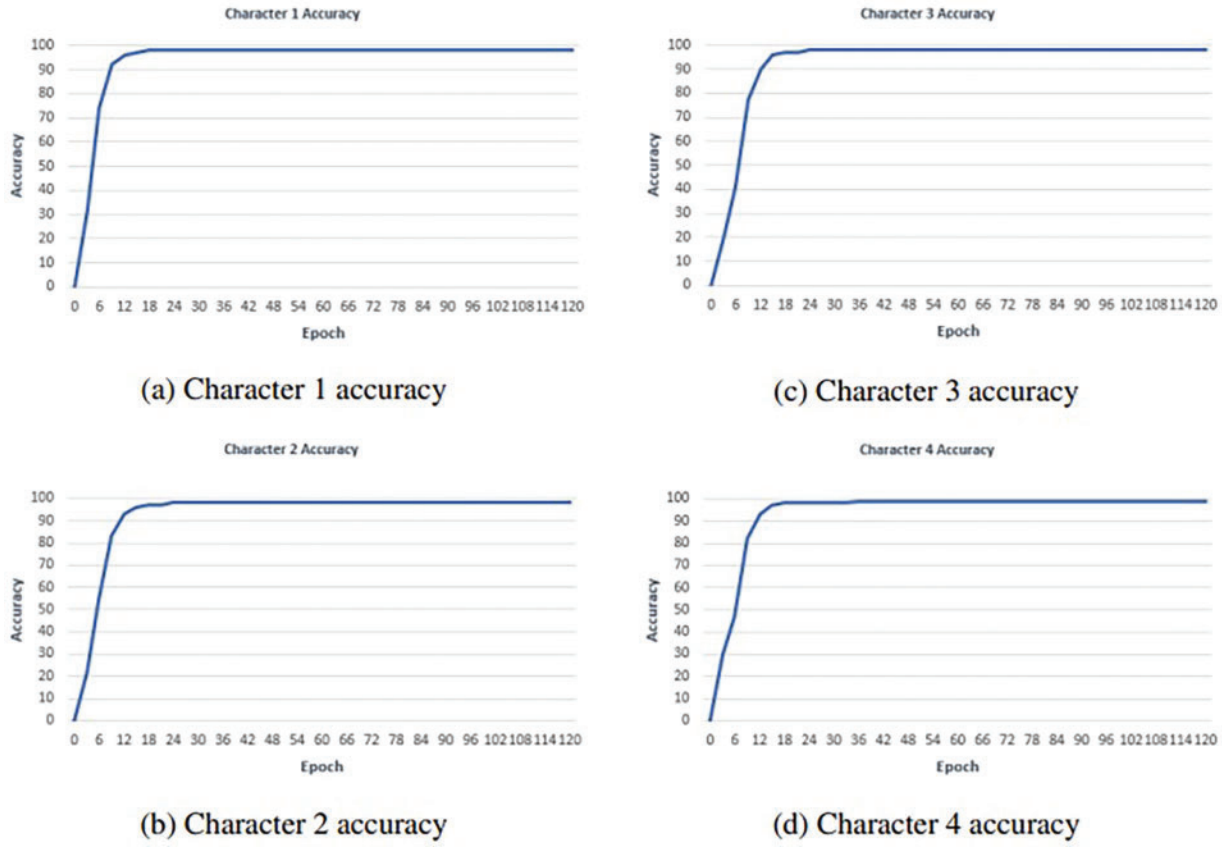

Figure 9: Character accuracy for Characters 1 (a), 2 (b), 3 (c) and 4 (d) using the DLACRM model on the Weibo dataset during the training phase

The training performance of the model on the Captcha 0.3 dataset, focusing on total and overall accuracy metrics, shows a clear trend of improvement throughout the 120 training epochs, as presented in Fig. 10. Initially, both total and overall accuracies were at 0% at epoch 0, reflecting the starting point with no initial learning. By epoch 3, the total accuracy had increased to 23%, while the overall accuracy remained at 0%, indicating an early stage of learning with limited generalization. As training progressed, the total accuracy showed substantial improvement, reaching 48% by epoch 6 and climbing to 70% by epoch 9. By epoch 12, the total accuracy stabilized at 82%, highlighting the model's rapid learning capabilities. From mid-training onward, the total accuracy consistently maintained high levels, indicating strong proficiency in recognizing characters accurately. The overall accuracy, while improving more gradually, reached 15% by epoch 6 and continued to rise, achieving 44% by epoch 9 and stabilizing at 63% by epoch 12. This consistent improvement underscores the model's ability to generalize effectively across different characters and tasks as training progressed, highlighting robust learning and adaptability.

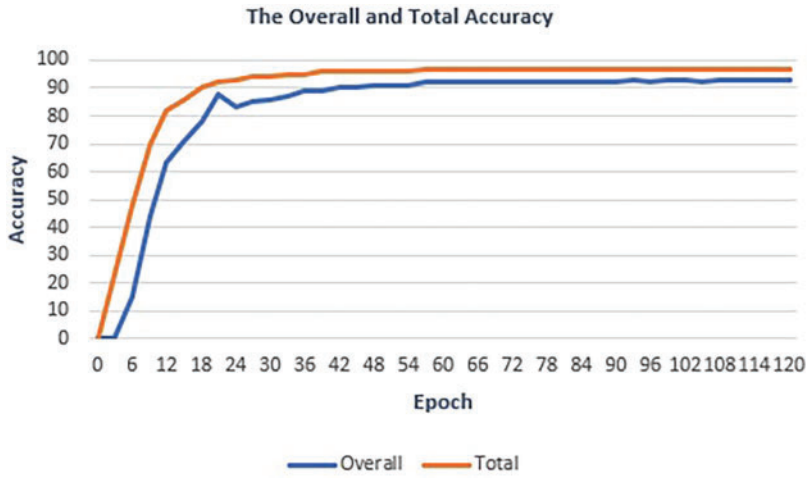

Figure 10: The overall and total character recognition accuracy of DLACRM on the Captcha 0.3 dataset during the training phase

Fig. 11 provides the training performance of the DLACRM model on the Captcha 0.3 dataset, with a specific focus on the accuracy for Characters 1, 2, 3, and 4 during the 120 epochs. Character 1 demonstrated a gradual improvement during training, starting at 0% accuracy at epoch 0 and reaching 17% by epoch 3. By epoch 6, Character 1's accuracy had increased significantly to 59%, and by epoch 9, it reached 82%, stabilizing at 90% from epoch 12 onwards, indicating strong learning and consistent performance. Character 2 started with a lower initial accuracy of 0%, reaching 15% by epoch 3. Despite a slower start, Character 2 showed steady progress, achieving 43% by epoch 6 and 69% by epoch 9, before stabilizing at 80% from epoch 12, reflecting effective adaptation even with variability early on.

Character 3 had a challenging start, beginning at 0% and improving to 17% by epoch 3. However, accuracy increased steadily to 39% by epoch 6 and 61% by epoch 9, stabilizing at 76% from epoch 12 onwards, demonstrating consistent growth and adaptation throughout the training. Character 4 followed a slightly faster learning curve, starting at 0% and improving to 41% by epoch 3. By epoch 6, Character 4 reached 51%, and by epoch 9, it climbed to 69%, stabilizing at 82% from epoch 12 onwards, highlighting robust learning capabilities across diverse characters. Overall, each character experienced substantial improvements within the initial epochs, with stabilization at high accuracy levels (76%–90%) by epoch 12. This consistent performance highlights the model's robust learning capabilities and its effectiveness in character recognition tasks across the dataset.

The overall and total accuracy of the model on the Gregwar dataset shows a gradual improvement over 120 epochs, as presented in Fig. 12. Initially, both accuracies were at 0%, with the total accuracy increasing to 10% by epoch 3 and 35% by epoch 12. By epoch 33, the total accuracy reached 65%, continuing to improve to 69% at epoch 45, and stabilizing at 73% from epochs 69 to 84. The overall accuracy, which started improving later, reached 2% by epoch 12, then climbed to 23% by epoch 33, 30% by epoch 45, and stabilized at 35-36% by epochs 72 and 84. These results reflect steady, consistent learning with slow but continuous gains in performance across tasks.

Fig. 13 provides the training performance of the DLACRM model on the Gregwar dataset, with a specific focus on the accuracy for Characters 1, 2, 3, and 4 during the 120 epochs. Character 1 demonstrated gradual learning progress during training, starting at 0% accuracy at epoch 0 and improving slowly to 6% by epoch 3. By epoch 6, Character 1's accuracy increased to 16%, and by epoch 9, it reached 28%, continuing to

improve to 39% by epoch 12. By epoch 33, Character 1 had improved significantly to 69%, showing a marked increase in performance. At epochs 45 and 72, the accuracy of Character 1 rose to 74% and 77%, respectively, maintaining stability at 77% by epoch 84, reflecting consistent learning and strong adaptation. Character 2 started similarly at 0% and reached 7% by epoch 3. Its accuracy improved gradually to 12% by epoch 6, 21% by epoch 9, and stabilized at 28% by epoch 12. By epoch 33, Character 2 reached 59% and continued its upward trend to 64% by epoch 45 and 68% by epoch 72, finally stabilizing at 69% by epoch 84, highlighting steady improvement across the training period.

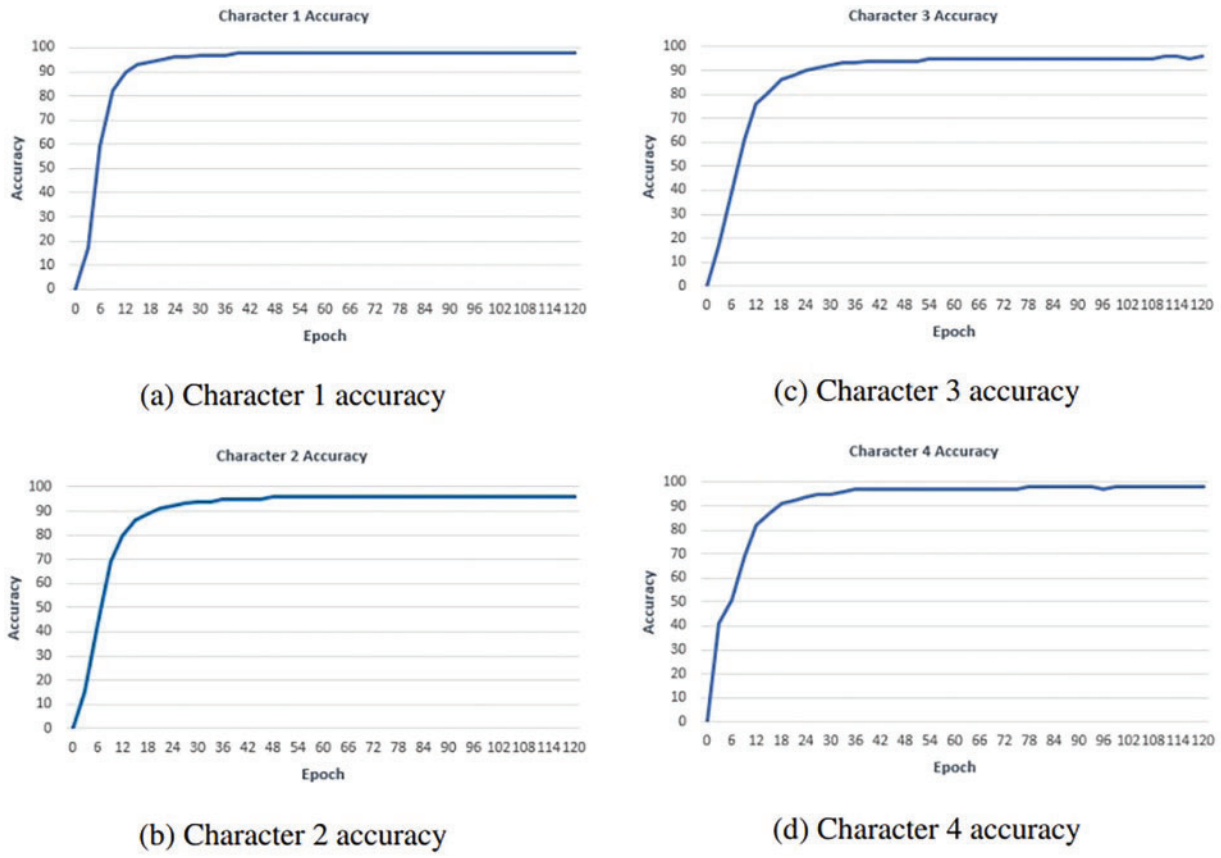

Figure 11: Character accuracy for Characters 1 (a), 2 (b), 3 (c) and 4 (d) using the DLACRM model on the Captcha 0.3 dataset during the training phase

Character 3 had a challenging start, beginning at 0% and reaching 9% by epoch 3. Accuracy increased slowly to 13% by epoch 6 and 18% by epoch 9, then improved to 24% by epoch 12. By epoch 33, Character 3 achieved 56% accuracy, reaching 62% by epoch 45 and 67% by epoch 72, and stabilizing at 67% by epoch 84, showing consistent but gradual learning. Character 4 followed a slightly faster learning curve, starting at 0% and reaching 18% by epoch 3. By epoch 6, Character 4 improved to 37%, reached 44% by epoch 9 and stabilized at 47% by epoch 12. At epoch 33, Character 4's accuracy climbed to 74%, increasing to 77% by epoch 45 and stabilizing at 80% by epochs 72 and 84, indicating a relatively quicker adaptation and robust performance. Overall, each character experienced substantial improvements within the initial and later epochs, with stabilization at higher accuracy levels (67%–80%) by epochs 72 and 84. This consistent yet modest progress reflects the model's gradual learning capabilities on the Gregwar dataset, demonstrating steady increases in character recognition skills over time.

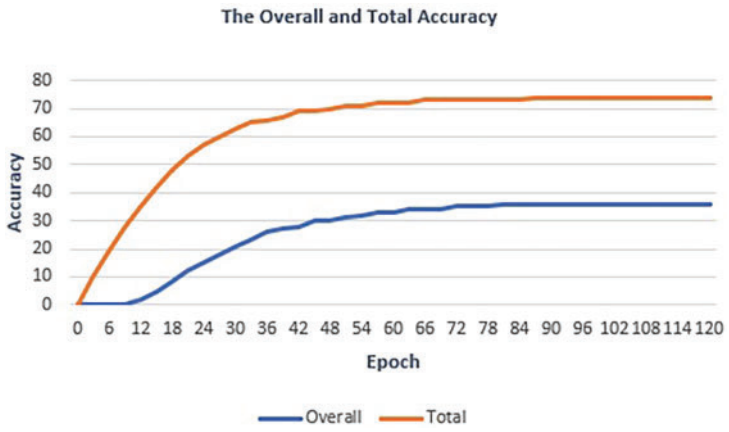

Figure 12: The overall character recognition accuracy of DLACRM on the Gregwar dataset during the training phase

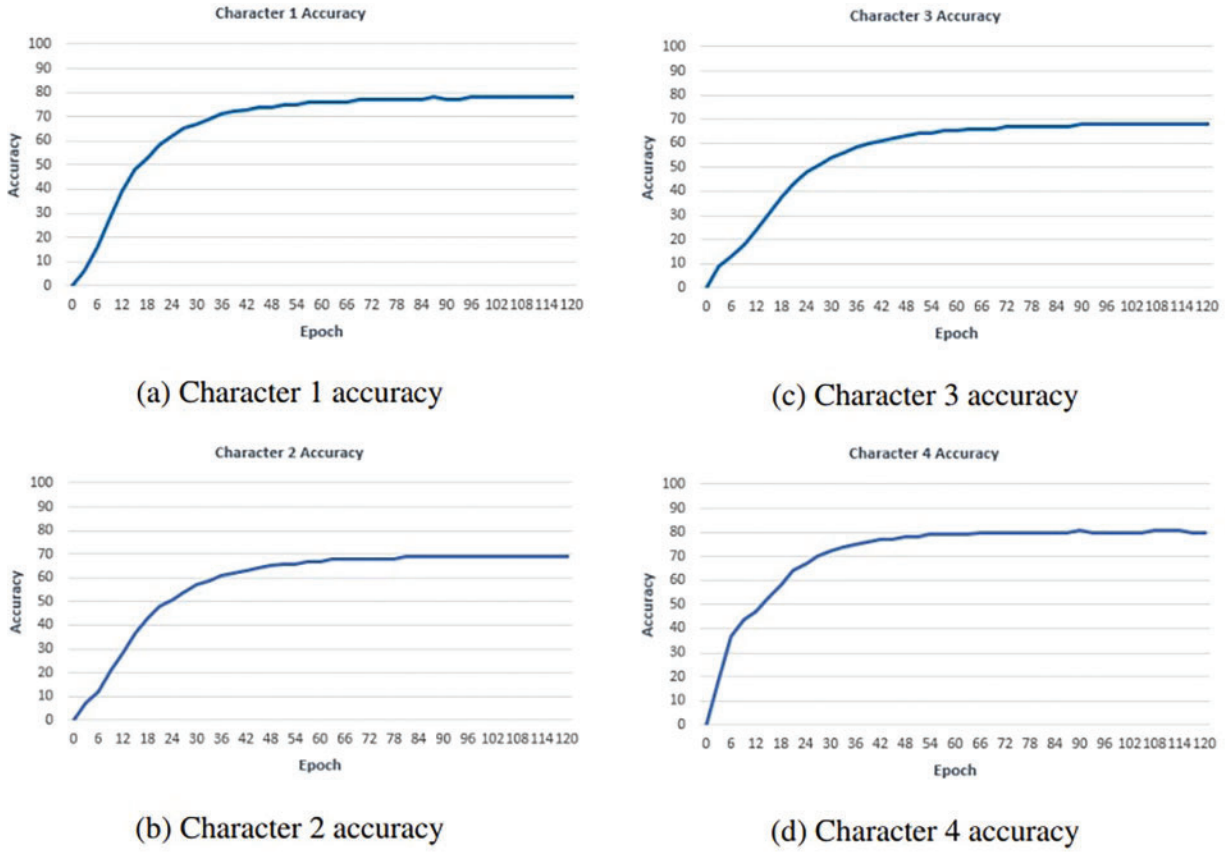

Figure 13: Character accuracy for Characters 1 (a), 2 (b), 3 (c) and 4 (d) using the DLACRM model on the Gregwar dataset during the training phase

4.4 Comparison Results

To thoroughly evaluate the strengths and weaknesses of the DLACRM model, we conducted an extensive comparison with several widely recognized CAPTCHA recognition algorithms, ensuring a fair assessment by using the same datasets. Key models in the image-text CAPTCHA recognition domain

included in our comparison were the Multilabel CNN, CRABI, and CRNN models. The Multilabel CNN architecture employs a single convolutional neural network with multiple softmax output layers, each tasked with recognizing a different character in the CAPTCHA. In contrast, CRABI simplifies the process by eliminating the need for segmentation, utilizing binary images attached to CAPTCHA copies and a basic CNN with a single softmax output layer for efficient character recognition, while the CRNN model combines convolutional layers with recurrent layers, resulting in a complex and resource-intensive architecture.

The detailed comparison results, presented in Tables 3–6, along with Figs. 14–17, which correspond to the BoC, Weibo, Captcha 0.3, and Gregwar CAPTCHA schemes, respectively, provide a comprehensive analysis of each model's performance across diverse CAPTCHA scenarios. The evaluation emphasizes two primary metrics: total character accuracy and overall CAPTCHA accuracy, offering valuable insights into the effectiveness of the tested models. Interestingly, the CRABI model outperformed all others in the Gregwar CAPTCHA scheme, including the DLACRM model, emphasizing the diversity and complexity of CAPTCHA structures and highlighting the need for recognition algorithms tailored to specific CAPTCHA types. Further analysis of the results in Tables 3 and 4 reveals that the DLACRM model outperformed all other methods in both total character accuracy and overall CAPTCHA accuracy for the BoC and Weibo CAPTCHA schemes, demonstrating its effectiveness in accurately recognizing characters and capturing the overall context of CAPTCHA images. Although the DLACRM model achieved the highest rankings in the BoC and Weibo schemes, it still performed admirably in the Captcha 0.3 scheme, ranking second behind the Multilabel model, which underscores the nuanced nature of CAPTCHA challenges and the necessity for adaptive models to maintain consistent efficacy across different CAPTCHA designs. Overall, the comparison results demonstrate the versatility and robustness of the DLACRM model, which shows superior performance across multiple CAPTCHA schemes, indicating its potential for widespread adoption in real-world applications where accurate and efficient CAPTCHA recognition is essential.

Table 3: Performance comparison results employing the BoC CAPTCHA scheme

Metric	Multilabel [35]	CRABI [35]	CRNN [35]	DLACRM
Total character accuracy testing	99.03% (39,614/40,000)	98.44% (39,379/40,000)	–	99.06% (39,627/40,000)
Overall CAPTCHA accuracy testing	96.39% (9639/10,000)	94.33% (9433/10,000)	96.47% (9647/10,000)	96.70% (9670/10,000)

Table 4: Performance comparison results employing the weibo CAPTCHA scheme

Metric	Multilabel [34]	CRABI [34]	CRNN [34]	DLACRM
Total character accuracy testing	96.03% (38,411/40,000)	97.89% (39,156/40,000)	–	98.72% (39,490/40,000)
Overall CAPTCHA accuracy testing	86.24% (8624/10,000)	92.68% (9268/10,000)	91.05% (9105/10,000)	95.92% (9592/10,000)

Table 5: Performance comparison results employing the Captcha 0.3 CAPTCHA scheme

Metric	Multilabel [35]	CRABI [35]	CRNN [35]	DLACRM
Total character accuracy testing	98.71% (39,485/40,000)	96.11% (38,444/40,000)	–	97.78% (39,115/40,000)
Overall CAPTCHA accuracy testing	95.33% (9533/10,000)	85.93% (8593/10,000)	83.57% (8357/10,000)	93.76% (9376/10,000)

Table 6: Performance comparison results employing the Gregwar CAPTCHA scheme

Metric	Multilabel [34]	CRABI [34]	CRNN [34]	DLACRM
Total character accuracy testing	83.31% (33,322/40,000)	85.28% (34,111/40,000)	–	76.24% (30,499/40,000)
Overall CAPTCHA accuracy testing	51.23% (5123/10,000)	54.20% (5420/10,000)	49.98% (4998/10,000)	40.57% (4057/10,000)

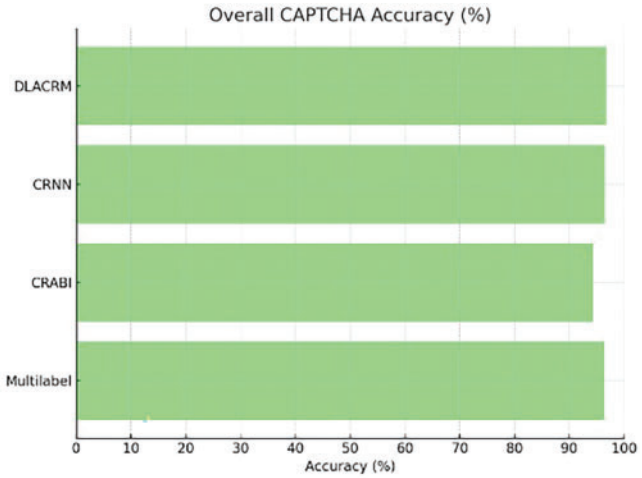

Figure 14: Compares the overall CAPTCHA accuracy of the DLACRM model with several other models, based on the BoC scheme

4.5 Qualitative Evaluation

To assess the effectiveness of the CAPTCHAs produced by DLACRM, it's crucial to complement quantitative score analysis with a qualitative evaluation. Fig. 18 presents a carefully selected array of sample images from the test dataset, each linked to its corresponding CAPTCHA. In this figure, every image is associated with CAPTCHA text generated by our model across four datasets: BoC, Weibo, Captcha 0.3, and Gregwar. For example, the image located in the top-left corner of the first row and column illustrates the BoC CAPTCHA "P6DF," while the top-right image represents the Gregwar CAPTCHA "k5Su." Our DLACRM model consistently demonstrates its ability to accurately capture the textual content within these images. The

model's performance and the quality of the CAPTCHAs it generates remain consistently high, as reflected in the scores shown in Table 2 and the example CAPTCHAs depicted in Fig. 18.

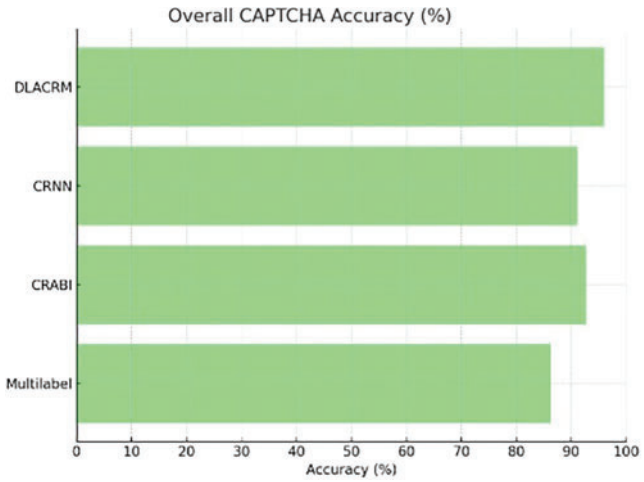

Figure 15: Compares the overall CAPTCHA accuracy of the DLACRM model with several other models, based on the Weibo scheme

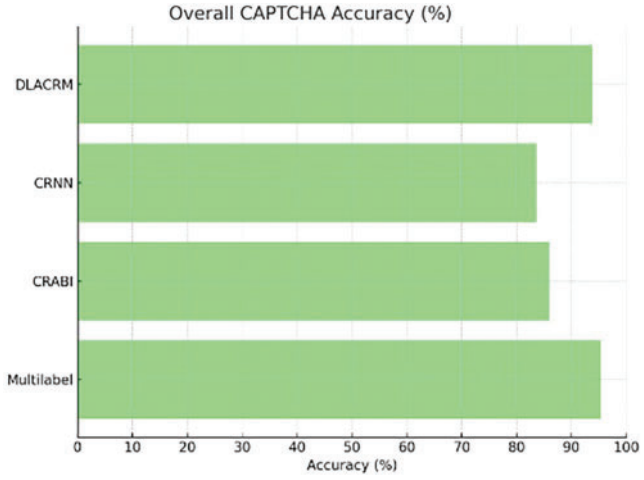

Figure 16: Compares the overall CAPTCHA accuracy of the DLACRM model with several other models, based on the Captcha 0.3 scheme

Fig. 19 highlights instances where CAPTCHAs were incorrectly recognized. For example, in the left CAPTCHA image, the character "l" was misidentified, and in the right image, the characters "g" and "j" were also incorrectly interpreted. These recognition errors are primarily due to overlapping colored lines that intersect the text and the challenging arrangement of the characters. The common failure scenarios typically stem from various anti-recognition mechanisms designed to thwart automated solvers. These mechanisms include the use of distorted and stretched characters, colored lines that overlay the text, and added background noise, all of which contribute to the increased complexity of the CAPTCHA. Additionally, the close spacing and occasional overlap of characters complicate the segmentation process, while the low contrast between the text and background further challenges machine-based CAPTCHA solvers.

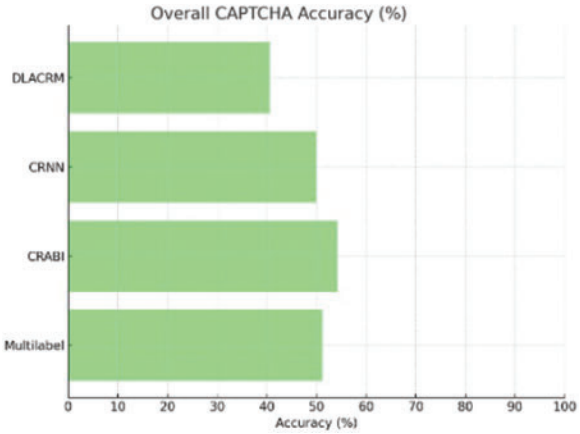

Figure 17: Compares the overall CAPTCHA accuracy of the DLACRM model with several other models, based on the Gregwar scheme

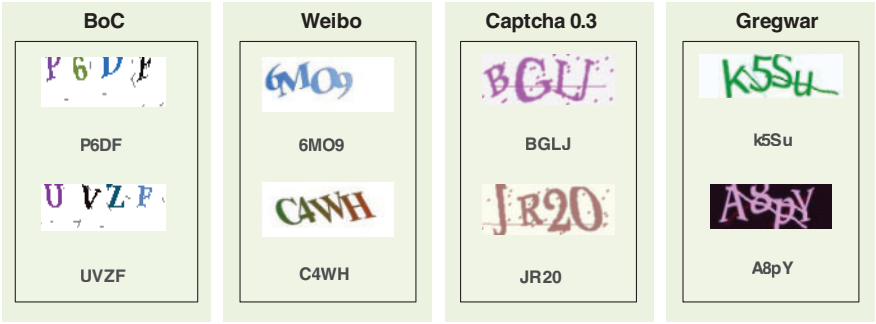

Figure 18: Examples of some CAPTCHAs correctly recognized by DLACRM for the four datasets: BoC, Weibo, Captcha 0.3, and Gregwar

Figure 19: Examples of CAPTCHAs incorrectly recognized by the DLACRM model, with "I" referring to the incorrectly recognized characters and "G" representing the ground truth

4.6 Discussion

This study introduces a groundbreaking approach specifically designed for the complex task of CAPTCHA recognition, marking a significant leap forward in the field. Our model integrates a two-layer attention mechanism—comprising CVA and GVA—that enhances its ability to capture relevant local visual features and accurately identify CAPTCHA characters. Through thorough evaluation and rigorous testing, our algorithm has demonstrated outstanding performance across various challenging CAPTCHA defense mechanisms. Our comprehensive assessment reveals the algorithm's robustness in overcoming complex challenges inherent in CAPTCHA designs, such as overlapping characters, noise lines, rotations, distortions, and varied color backgrounds. Additionally, the algorithm's adaptability in handling multiple CAPTCHA character categories further highlights its versatility and wide-ranging applicability.

Notably, our algorithm exhibits remarkable resilience when facing the formidable defense mechanisms of the Gregwar CAPTCHA scheme, which is known for its stringent security measures. Traditional CAPTCHA recognition methods often struggle with this scheme, but our algorithm rises to the challenge, delivering exceptional accuracy and effectiveness in deciphering even the most intricate Gregwar CAPTCHAs. Furthermore, our approach streamlines the recognition process by eliminating the need for the cumbersome task of segmenting CAPTCHA images into individual characters, which significantly improves both accuracy and efficiency. By pushing the boundaries of CAPTCHA recognition technology, our work not only advances the state-of-the-art but also has significant implications for enhancing online security measures. The robustness and versatility of our algorithm pave the way for more secure online platforms, protecting against automated threats and ensuring the integrity of digital interactions.

Compared to the CRNN model, which involves numerous hyperparameters that require careful tuning and convolutional layers that need specific adjustments, our DLACRM model offers a much simpler and more efficient architecture. The complexity of CRNN's design adds significant challenges, especially when optimizing the model for effective performance. In contrast, our DLACRM model features relatively few hyperparameters and a flexible CNN architecture that integrates seamlessly with the RNN layer. This design allows for efficient processing without the need for extensive preprocessing steps. Additionally, our model does not depend on the number of characters in the image, which eliminates the limitations faced by other models like CRABI that require complex preprocessing, longer training times, and complicated architectures. While the CRABI model, which utilizes Attached Binary Images (ABIs) technology, suffers from the need to process multiple copies of input images sequentially, our DLACRM model bypasses these challenges. Also, our model does not suffer from the scalability issues when the number of characters in CAPTCHA images increases as compared to multilabel CNN approach. A key strength of our approach is its novelty; we have pioneered the use of image captioning models for CAPTCHA recognition, which enhances both the performance and accuracy of CAPTCHA recognition systems. Our approach can be extended to cybersecurity domains that rely on sequential data. The advanced attention mechanisms, especially in processing such data, have proven effective and hold significant potential for enhancing cybersecurity measures.

5 Conclusions

In this study, we introduce an innovative two-layer attention framework for CAPTCHA recognition that builds on traditional attention mechanisms by incorporating Guided Visual Attention (GVA) to sharpen focus on key features. We have adapted the well-established image captioning task to meet this specific need. Our approach features the GVA module guided by the Conventional Visual Attention (CVA) component, integrating two LSTM layers to enhance recognition of CAPTCHAs. Our system demonstrates impressive success across diverse datasets, including BoC, Weibo, Gregwar, and Captcha 0.3, achieving high performance without the need for segmentation. This research not only advances theoretical understanding

but also has practical implications for strengthening online security. The simplicity and versatility of our model make it a promising tool for CAPTCHA recognition, utilizing techniques from image captioning. This work marks a significant advancement in CAPTCHA security, highlighting the potential of image captioning techniques to improve internet security. In future work, we aim to enhance our approach by incorporating Transformer models, leveraging self-attention mechanisms, and employing multi-head attention to improve performance and scalability.

Acknowledgement: None.

Funding Statement: This work is supported by the National Natural Science Foundation of China (Nos. U22A2034, 62177047), High Caliber Foreign Experts Introduction Plan funded by MOST, and Central South University Research Programme of Advanced Interdisciplinary Studies (No. 2023QYJC020).

Author Contributions: Zaid Derea was responsible for investigation, conceptualization, and software development. Xiaoyan Kui contributed to the review and editing of the manuscript. Beiji Zou provided supervision throughout the project. Alaa Thobhani handled conceptualization, validation, and visualization, while Amr Abdussalam contributed resources. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: Due to privacy concerns and proprietary restrictions, the dataset cannot be shared openly. The data were sourced from publicly available platforms, but the specific dataset remains confidential to protect participant privacy and ownership rights.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest to report regarding the present study.

Novelty Statement

This work introduces a novel CAPTCHA recognition framework that leverages a two-layer visual attention mechanism, marking a significant departure from traditional approaches. Unlike existing models that rely on single-layer attention or omit attention layers altogether, our method incorporates Guided Visual Attention (GVA) to dynamically refine feature extraction and enhance recognition accuracy. By adapting advanced image captioning techniques, the framework integrates a tailored UpDown model with dual LSTM layers, addressing challenges like character overlap and distortion. Extensive evaluation across diverse datasets, including real-world CAPTCHAs, demonstrates superior accuracy and robustness, achieving state-of-the-art results. This innovative approach not only advances CAPTCHA recognition but also provides a versatile solution for bolstering online security against automated threats.

References

1. Sinha S, Surve MI. CAPTCHA recognition and analysis using custom based CNN Model-capsecure. In: 2023 International Conference on Emerging Techniques in Computational Intelligence (ICETCI); 2023; IEEE. p. 244–50.
2. Kumar M, Jindal M, Kumar M. An efficient technique for breaking of coloured Hindi CAPTCHA. Soft Comput. 2023;27(16):11661–86. doi:10.1007/s00500-023-07844-3.
3. Wang P, Gao H, Guo X, Xiao C, Qi F, Yan Z. An experimental investigation of text-based CAPTCHA attacks and their robustness. ACM Comput Surv. 2023;55(9):1–38. doi:10.1145/3559754.
4. Hussain R, Gao H, Shaikh RA, Soomro SP. Recognition based segmentation of connected characters in text based CAPTCHAs. In: 2016 8th IEEE International Conference on Communication Software and Networks (ICCSN); 2016; IEEE. p. 673–6.
5. Von Ahn L, Blum M, Hopper NJ, Langford J. CAPTCHA: using hard AI problems for security. In: Eurocrypt, Springer; 2003. Vol. 2656, p. 294–311.

6. Von Ahn L, Blum M, Langford J. Telling humans and computers apart automatically. Commun ACM. 2004;47(2):56–60. doi:10.1145/966389.966390.
7. Kumar M, Jindal M, Kumar M. A systematic survey on CAPTCHA recognition: types, creation and breaking techniques. Arch Comput Methods Eng. 2022;29(2):1107–36. doi:10.1007/s11831-021-09608-4.
8. Tang M, Gao H, Zhang Y, Liu Y, Zhang P, Wang P. Research on deep learning techniques in breaking text-based captchas and designing image-based captcha. IEEE Trans Inf Forensics Secur. 2018;13(10):2522–37. doi:10.1109/TIFS.2018.2821096.
9. Wang P, Gao H, Rao Q, Luo S, Yuan Z, Shi Z. A security analysis of captchas with large character sets. IEEE Trans Dependable Secure Comput. 2020;18(6):2953–68. doi:10.1109/TDSC.2020.2971477.
10. Ye G, Tang Z, Fang D, Zhu Z, Feng Y, Xu P, et al. Yet another text captcha solver: a generative adversarial network based approach. In: Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security; 2018. p. 332–48.
11. Gao H, Wang W, Qi J, Wang X, Liu X, Yan J. The robustness of hollow CAPTCHAs. In: Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security; 2013. p. 1075–86.
12. Malik S, Soundararajan R. Llrnet: a multiscale subband learning approach for low light image restoration. In: 2019 IEEE International Conference on Image Processing (ICIP); 2019; IEEE. p. 779–83.
13. Jin Z, Iqbal MZ, Bobkov D, Zou W, Li X, Steinbach E. A flexible deep CNN framework for image restoration. IEEE Trans Multimedia. 2019;22(4):1055–68. doi:10.1109/TMM.2019.2938340.
14. Dong W, Wang P, Yin W, Shi G, Wu F, Lu X. Denoising prior driven deep neural network for image restoration. IEEE Trans Pattern Anal Mach Intell. 2018;41(10):2305–18. doi:10.1109/TPAMI.2018.2873610.
15. Liu Y. An improved faster R-CNN for object detection. In: 2018 11th Int Symp Comput Intell Des (ISCID); 2018: IEEE. Vol. 2, p. 119–23. doi:10.1109/ISCID.2018.10128
16. Abdussalam A, Sun S, Fu M, Sun H, Khan I. License plate segmentation method using deep learning techniques. In: Signal and Information Processing, Networking and Computers: Proceedings of the 4th International Conference on Signal and Information Processing, Networking and Computers (ICSINC); 2019; Springer. p. 58–65.
17. Abdussalam A, Sun S, Fu M, Ullah Y, Ali S. Robust model for chinese license plate character recognition using deep learning techniques. In: Communications, Signal Processing, and Systems: Proceedings of the 2018 CSPS; 2020, Springer. p. 121–7.
18. Anderson P, He X, Buehler C, Teney D, Johnson M, Gould S, et al. Bottom-up and top-down attention for image captioning and visual question answering. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; 2018. p. 6077–86.
19. Hossen MB, Ye Z, Abdussalam A, Hossain MI. GVA: guided visual attention approach for automatic image caption generation. Multimed Syst. 2024;30(1):50. doi:10.1007/s00530-023-01249-w.
20. Shi C, Ji S, Liu Q, Liu C, Chen Y, He Y, et al. Text captcha is dead? a large scale deployment and empirical study. In: Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security; 2020. p. 1391–406.
21. Wan X, Johari J, Ruslan FA. Adaptive CAPTCHA: a CRNN-based text CAPTCHA solver with adaptive fusion filter networks. Appl Sci. 2024;14(12):5016. doi:10.3390/app14125016.
22. Zhang N, Ebrahimi M, Li W, Chen H. A generative adversarial learning framework for breaking text-based captcha in the dark web. In: 2020 IEEE International Conference on Intelligence and Security Informatics (ISI); 2020; IEEE. p. 1–6.
23. Tariq N, Khan FA, Moqurrab SA, Srivastava G. CAPTCHA types and breaking techniques: design issues, challenges, and future research directions. arXiv:230710239. 2023.
24. Chow YW, Susilo W, Thorncharoensri P. CAPTCHA design and security issues. In: Advances in cyber security: principles, techniques, and applications. Springer; 2019. p. 69–92. doi:10.1007/978-981-13-1483-4_4.
25. Guerar M, Verderame L, Migliardi M, Palmieri F, Merlo A. Gotta CAPTCHA' Em all: a survey of 20 Years of the human-or-computer Dilemma. ACM Comput Surv. 2021;54(9):1–33. doi:10.1145/3477142.
26. Wang Q, Ibrahim S, Idrus Z. A systematic literature review: integrating deep learning models for visual-based CAPTCHA generation. J Autono Intell. 2024;7(5):1–15. doi:10.32629/jai.v7i5.1551.

27. Sharma S, Singh D. CAPTCHA in web security and deep-captcha configuration based on machine learning. In: 2024 3rd International Conference for Innovation in Technology (INOCON); 2024; IEEE. p. 1–6.
28. Algwil AM. A security analysis of text-based captcha schemes. African J Adv Pure Appld Sci (AJAPAS). 2023;2(3):309–23.
29. Xu Z, Yan Q. Boosting the transferability of adversarial CAPTCHAs. Comput Secur. 2024;145(7):104000. doi:10.1016/j.cose.2024.104000.
30. Ravi V, Alazab M, Soman K, Srinivasan S, Venkatraman S, Pham QV, et al. Deep learning for cyber security applications: a comprehensive survey. Comput Process. 2021. doi:10.36227/techrxiv.16748161.v1.
31. Bursztein E, Martin M, Mitchell J. Text-based CAPTCHA strengths and weaknesses. In: Proceedings of the 18th ACM Conference on Computer and Communications Security; 2011. p. 125–38.
32. Shao R, Shi Z, Yi J, Chen PY, Hsieh CJ. Robust text captchas using adversarial examples. In: 2022 IEEE International Conference on Big Data (Big Data); 2022; IEEE. p. 1495–504.
33. Weng H, Zhao B, Ji S, Chen J, Wang T, He Q, et al. Towards understanding the security of modern image captchas and underground captcha-solving services. Big Data Min Analy. 2019;2(2):118–44. doi:10.26599/BDMA.2019.9020001.
34. Thobhani A, Gao M, Hawbani A, Ali STM, Abdussalam A. CAPTCHA recognition using deep learning with attached binary images. Electronics. 2020;9(9):1522. doi:10.3390/electronics9091522.
35. Derea Z, Zou B, Al-Shargabi AA, Thobhani A, Abdussalam A. Deep learning based CAPTCHA recognition network with grouping strategy. Sensors. 2023;23(23):9487. doi:10.3390/s23239487.
36. Khatavkar V, Velankar M, Petkar S. Segmentation-free connectionist temporal classification loss based OCR model for text captcha classification. arXiv:240205417. 2024.
37. Chang G, Gao H, Pei G, Luo S, Zhang Y, Cheng N, et al. The robustness of behavior-verification-based slider CAPTCHAs. J Inf Secur Appl. 2024;81(2):103711. doi:10.1016/j.jisa.2024.103711.
38. Adesina AO, Ayobioloja PS, Obagbuwa IC, Odule TJ, Afolorunso AA, Ajagbe SA. An improved text-based and image-based CAPTCHA based on solving and response time. Comput Mater Contin. 2023;74(2):2661–75. doi:10.32604/cmc.2023.031245.
39. Thobhani A, Zou B, Kui X, Al-Shargabi AA, Derea Z, Abdussalam A, et al. A novel image captioning model with visual-semantic similarities and visual representations re-weighting. J King Saud Univ-Comput Inf Sci. 2024;36(7):102127. doi:10.1016/j.jksuci.2024.102127.
40. Thobhani A, Zou B, Kui X, Abdussalam A, Asim M, Ahmed N, et al. A concise and varied visual features-based image captioning model with visual selection. Comput Mater Contin. 2024;81(2):2873–94. doi:10.32604/cmc.2024.054841.