International Journal of Machine Learning and Cybernetics (2023) 14:959–972
https://doi.org/10.1007/s13042-022-01675-8

ORIGINAL ARTICLE

Multiview deep learning-based attack to break text-CAPTCHAs

Mukhtar Opeyemi Yusuf[1] · Divya Srivastava[1] · Deepak Singh[2] · Vijaypal Singh Rathor[3]

Received: 17 December 2021 / Accepted: 22 September 2022 / Published online: 3 October 2022

Abstract

Completely Automated Public Turing Test To Tell Computer and Humans Apart (CAPTCHA) is a computer program that prevents malicious computer users. Text-CAPTCHA schemes utilize less-computational costs. Hence, they are the most popularly used. This paper investigates the effectiveness of state-of-the-art (SOTA) text-CAPTCHA schemes, proposes a Multiview deep learning system to break them, and highlights their weaknesses. Rather than the usual single-view feature extraction, the proposed model explores correlational features from multiple views to increase the model's generalization and classification accuracy. The model combines convolutional neural networks and recurrent networks to preserve the input text-CAPTCHA's spatial and sequential order. The proposed system has successfully achieved average accuracies ranging from 93.6% to 100%, and the average time to break a text-CAPTCHA scheme ranges from 0.0032 to 0.21 seconds on eight different datasets. Furthermore, an ablation study on 71 human users was conducted to evaluate the effectiveness of the schemes. The results demonstrated that the proposed system effectively outperforms the human users whom the schemes were designed to serve. Lastly, when compared with existing systems, the proposed system outperforms existing SOTA systems with an accuracy gap of almost 40% higher.

Keywords CAPTCHA · Multiview learning classification · Multiview integration · Security and privacy · Discriminative features · Connectionist temporal classification

1 Introduction

With increasing interest in internet activities, protecting the authenticity and integrity of internet service users becomes more crucial. Over time, several approaches have been proposed that attempt to provide such protection. However, as technology advances, it exposes the weakness of such attempts. Completely Automated Public Turing Test To Tell Computer and Humans Apart (CAPTCHA) is one such attempt. CAPTCHA is a computer program that protects unauthorized bot(s) from disguising as an authorized user(s). The idea is quite simple—a computer program poses a relatable easy question for humans to solve. In contrast, the question must be unsolvable for computer programs (bots). Unlike the famous *Turing test* proposed by Alan Turing [33], the moderator, in this case, is a computer and not a human. Sometimes CAPTCHA programs are referred to as the *reversed Turing test*. The first CAPTCHA scheme was proposed by Von Ahn et al. [35]. Over time, CAPTCHA schemes are commercialized, for providing a simple but effective layer of security for services that trade data over the internet. Generally, CAPTCHA schemes tend to exploit the cognitive advantage of humans over computer systems. CAPTCHA schemes have been classified into two which are briefly discussed in this section.

Random CAPTCHA schemes: These kinds of schemes generate random tasks, easy for human users to solve, but are difficult for computer bots to solve. Over the years, these kinds of CAPTCHA schemes have evolved. They are broadly categorized into; text-based CAPTCHA, image-based CAPTCHA, video-based CAPTCHA, and audio-based CAPTCHA. The text-based CAPTCHA (or text-CAPTCHA) scheme generates random texts with some difficulty levels like rotation, warping, random noise, and so on. Such that malicious computer bot cannot identify the generated texts,

✉ Mukhtar Opeyemi Yusuf
e20soe814@bennett.edu.in

[1] Department of Computer Science and Engineering, Bennett University, Greater Noida, Utter Pradesh 201310, India

[2] Department of Computer Science and Engineering, National Institute of Technology, Raipur, India

[3] PDPM Indian Institute of Information Technology, Design and Manufacturing, Jabalpur, India

but, humans can. Image-CAPTCHA displays an image to a user, and the user is asked to identify or describe the image. Image-CAPTCHA is seen to be very powerful at the time and was adopted by small and big organizations. However, the implementation of such schemes can be computationally expensive compared to the text-CAPTCHA. More recently, powerful machine learning algorithms made it easier to defeat the Image-CAPTCHA scheme. An extension of the image-CAPTCHA scheme is the video-CAPTCHA. In this case, a video clip is played for a user to describe the content in the video, then answer some questions. A video clip is made up of thousands or millions of image frames. Hence, video-CAPTCHA will incur more computational cost than the image-CAPTCHA. Lastly, the audio-CAPTCHA schemes were introduced to mainly cater for users that are visually impaired. The user is asked a question in an audio, and the user is expected to provide answer to the question. Drop in network connectivity affects the quality of audio which makes it difficult for humans to solve.

Behavioural CAPTCHA: is the emerging trend in CAPTCHA schemes [42]. This kind of CAPTCHA scheme exploits different user parameters (like browsing history, location, cookies, and more) to classify a user's behavior without interacting directly with the user. This behavior classifies a user as either legitimate or a malicious computer bot. One advantage of such CAPTCHA schemes is that it saves the user the difficulty and time to solve CAPTCHA tests. However, this kind of CAPTCHA scheme sometimes fails to identify a user as legitimate if they choose to browse in incognito mode or when specific browser plugins block access to capture some parameters. A drawback of this scheme is that users cannot redeem themselves when the scheme has deemed a user illegitimate. Hence, denying access to the user's requested resource. An example of this kind of CAPTCHA scheme is the recent Google reCAPTCHA Enterprise.

The text-CAPTCHA is the most popular and widely applicable CAPTCHA scheme. The widespread application of this kind of scheme is because of its ease of implementation and low computational cost. Therefore, the work in this paper is confined to the text-CAPTCHA scheme. Text-CAPTCHA schemes are intuitively easy for humans to solve [6]. In designing any robust CAPTCHA scheme today, one must vividly understand the intuitive capabilities of humans and the current capabilities of computer systems [6]. The earliest form of the text-CAPTCHA scheme is the Altavista CAPTCHA in 1997 [4]. Altavista generates a simple string of distorted alphanumeric texts that is easy for humans to recognize but difficult for the computer programs of that time. With advancements in today's computer architectures and algorithms. It is feasible to deploy lightweight character recognition algorithms to break such a CAPTCHA scheme. Over time, researchers have introduced several security features to improve the security of text-CAPTCHA schemes, like in [7, 18]. The most commonly used security features are summarized as follows;

i *Isolated characters:* this feature evenly spaces out all characters from each other in the text-CAPTCHA. Figure 1a is a pictorial description of such technique.

ii *Overlapping:* this feature removes extra spaces and slightly overlaps each character. Such overlapping of characters increases the challenge for a character recognition bot. Figure 1b is a pictorial description of such technique.

iii *Rotation and random noise:* this feature randomly rotates the generated characters and adds some random noise(Gaussian, strokes, arcs, circles, and more) to confuse the bot's recognition algorithms. Figure 1c is a pictorial description of such technique.

iv *Warping:* this feature slightly distorts the edges and curves of each character. Such that the shape of each character is distorted yet understandable by humans. Figure 1d is a pictorial description of such technique.

v *Contour/Hollow enhancement:* this feature highlights the contour lines of each character. Such that line/edge connected forms hollow-shaped characters. Figure 1e is a pictorial description of such technique.

vi *Background as a noise:* this feature uses complicated and confusing background images that serve as noise to deceive a character recognition bot. Figure 1f is a pictorial description of such technique.

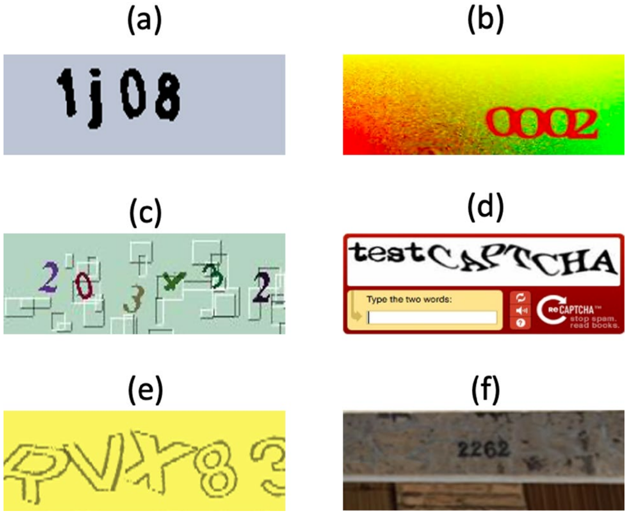


Fig. 1 Demonstrating some of the commonly used security features in securing text-CAPTCHA schemes: (**a**) Character isolated, (**b**) Overlapping, (**c**) Rotation and Random Noise, (**d**) Warping, (**e**) Hollow scheme, (**f**) Background as a noise

The organization of the rest of this paper is as follows. Section 2 highlights some related works. The methodology of the proposed system is discussed in Sect. 3. Experimental setup and discussion is presented in Sect. 4. Section 5 presents result discussion and analysis. Lastly, Sect. 6 summarized the conclusion and plans for future works.

2 Related works

The work presented in this paper is related to defeating text-CAPTCHA schemes. As they are widely applicable across the internet, both in the commercial and non-commercial sectors [14, 30]. Therefore, it is important to continue researching the security strengths of this kind of CAPTCHA scheme and bring awareness of its weaknesses to stakeholders. Defeating the text-based CAPTCHA scheme has been an open-end research for a while. This section discusses some of the existing works most related to the work proposed in this paper. Existing approaches were categorized as single-view learning and Multiview learning in this paper.

2.1 Single-view learning approach

Single-view learning approach indicates learning approaches that consider one viewpoint of the data. A view can be a measure or representation of data from the same modality, dimension, sources, features, and more. In a single-view learning setting, if the data consist of multiple views. They are concatenated to form a single view and then continue with training [40]. Several studies like [9, 25, 36, 45] have considered the single-view learning approach to defeat text-CAPTCHA schemes. This section discusses some of the single-view learning approaches related to the work described in this paper.

Rui et al. [29] proposed a 2-dimensional Recurrent Neural Network (2DRNN) that semantically segments each character in the text-CAPTCHA image. Such that it divides a static text-CAPTCHA image into different sequential timesteps. This idea succeeds the pre-segmentation process used in earlier works, like in [24]. The paper also implements a forward-backward connectionist temporal classification (CTC) function to align the sequential distribution of the output from the 2DRNN and return the probability of the corresponding text character. The authors further implement a genetic algorithm to optimize the predictions from the CTC.

More recently, [23] proposed a Neural CAPTCHA Network (NCN). The network utilizes convolutional blocks, followed by a fully connected (FC) layer. Then a bidirectional LSTM (BiLSTM) network is used to extract sequential information, and finally, they implement a CTC as the loss function and train the network end-to-end. This approach inspires the technique used in the proposed system. However, the proposed system further extends this idea to introduce Multiview representation of text-CAPTCHA data and encourages view collaboration during training to enhance the accuracy and generalization of the model. Zi et al. [45] proposed a deep convolutional neural network (CNN) to encode feature representations of the text-CAPTCHA image and an attention-based recurrent neural network (RNN) to decode the encoded features into textual characters. Nouri et al. [25] proposed a deep CNN architecture named Deep-CAPTCHA Network. It uses three convolutional and MaxPooling layers followed by an FC layer finally connected to $L - SoftMax$. L refers to the number of expected characters. A significant limitation to this work is that L is pre-defined before training, and therefore the model only works on fixed-length CAPTCHA schemes. Combining CNN with LSTM to extract spatial and sequential features is successful in other similar areas like image captioning [1].

Generative adversarial networks (GAN) have been very successful in many image processing task. Breaking text-CAPTCHAs is not an exception to this trend. In recent times, several GAN-based models are proposed to break text-CAPTCHA. Like in [19, 38, 41, 43, 44]. However, they all use very similar approaches. Firstly, a GAN model learns to generate synthetic text-CAPTCHA images. Therefore, the GAN inherently learns the security features embedded in the text-CAPTCHA image. Then the GAN model leverages this technique to de-noise the text-CAPTCHA image during preprocessing stage. A segmentation algorithm like the contour detection using border tracing [28], or CNN-based model to process the clean (denoised) text-CAPTCHA image. Finally, a recognition system identifies each pre-segmented character. Synonymously in all of these GAN-related works, the GANs serve as a preprocessing tool rather than an end-end model for breaking text-CAPTCHA. Unlike the work proposed in this paper, these GAN-related models still rely heavily on the segmentation algorithm and recognition systems to identify each character. The performance of the proposed system is evaluated and compared to some GAN related models.

2.2 Multiview learning approach

The Multiview learning approach explores multiple view data to learn discriminative features from different views to improve the model's generalization. Conventional machine/deep learning systems mostly use a single-view learning technique. Single-view learning assumes that dominant features within the data is enough for the model to make conclusions. However, this is not always the case in a real-world scenario. For example, in medical diagnosis, several kinds of medical data are required, from clinical data (patient medical history) to different lab-tests (blood test, blood pressure, sugar level), to medical imaging (X-RAY, MRI, CT-Scan, and many more). These data are from different sources and modalities, and wholistically, they provide a proper indication of the patient's diagnosis. With single-view learning setting, the model is trained to maximize the likelihood of a patient diagnosis from the combined multiple views. This brute force approach does not properly consider the complex relationship within the Multiview data. Using the same example, if features from the lab-test view are highly dominant over the remaining views. The model could diagnose *pulmonary tuberculosis*, whereas, with proper consideration of the medical history and imaging, *lung cancer* could be detected. In such a scenario, a Multiview learning system aspires to systematically learn discriminative and mutual information from each view to provide a well informed diagnostic inference.

There are three different techniques for integrating views in Multiview learning systems. These are; early, intermediate, and late integration [34]. In early integration, the views are fused/concatenated together at the beginning of the model's architecture like in [13, 22]. In intermediate integration, features are extracted from each view separately, then later integrated sometime during the training/learning process, like in [10, 11]. Late integration trains each view separately. Then a different learner (also referred to as the base learner) optimizes the outcome from each trained view. Such kind of integration technique is more useful when there exist incomplete views or view disagreement amongst the views, like in [21, 31, 39].

So far, no existing work has explored the Multiview learning system to break text-CAPTCHA. However, two notable works have explored the Multiview learning technique in the area of CAPTCHA solutions, but they are unrelated to the approach proposed in this article. Like [15] proposed a Dynamically Weighted Multiview Semi-Supervised Learning approach to detect new attacking schemes by extracting hidden patterns from multiple perspectives and then updating each view's weight dynamically. In [27] a Multiview stacking ensemble learning system is proposed to optimize the computing power usage of endpoint devices and reduce the pressure on the cloud computing resources.

This paper proposes a novel Multiview deep learning system to break the existing text-based CAPTCHA scheme. The system is described in four stages. First is the synthesization of the text-CAPTCHA image to build a Multiview representation. The second stage implements a convolution block that extracts spatial features from the represented views. The third stage transformed the extracted spatial features into sequential features using a bidirectional gated recurrent unit (GRU). The final stage integrates the Multiview features and trains a classifier using the connectionist temporal classification (CTC) to align the output sequences to the expected text characters.

The main contribution of this paper is summarized as follows.

1 For the first time, introducing a Multiview learning approach to breaking text-CAPTCHA.
2 Proposal of a novel Multiview deep-learner architecture that extracts deep spatial and sequential features from multiple views, and encourages complementarity among the views. The model also performs on-the-fly classication/prediction of the given text-CAPTCHA.
3 An ablation study was conducted on eight different text-CAPTCHA schemes to measure the effectiveness of the text-CAPTCHA scheme on human users and to evaluate the proposed system's performance.
4 Extensive experiments demonstrate the effectiveness of the proposed system in terms of accuracy and execution time.

3 Proposed system

This section discusses the approach of the proposed system in detail. For clarity, the system is described in four stages; Multiview representation, Feature extraction, CAPTCHA character sequencing, View integration and classification/prediction. Meanwhile, a brief overview of the problem formulation is defined below.

3.1 Problem formulation

Assume $\vec{\mathbf{x}}^{\mathbf{a}} = [x_1^a, x_2^a, x_3^a, \dots x_N^a] \in \mathrm{IR}^{NXd_i}$, and $\vec{\mathbf{x}}^{\mathbf{b}} = [x_1^b, x_2^b, x_3^b, \dots x_N^b] \in \mathrm{IR}^{NXd_i}$ as two views of the same data named $\vec{\mathbf{X}} \in \mathrm{IR}^{2XN}$. Where N represent the number of samples in the dataset. d_i is the dimension of the i^{th} sample, x^{a_i} and x^{b_i} are the first and second data samples in each view a and b respectively. The proposed system extracts two kinds of deep-features. First, spatial features $Z_{sp}^{a_i}$ and $Z_{sp}^{b_i}$ are extracted from paired input views $\vec{\mathbf{X}}_{\mathbf{i}}^{\mathbf{a}}$ and $\vec{\mathbf{X}}_{\mathbf{i}}^{\mathbf{b}}$ respectively. Then sequential features $Z_{sq}^{a_i}$ and $Z_{sq}^{b_i}$ are extracted from $Z_{sp}^{a_i}$ and $Z_{sp}^{b_i}$ respectively. Features extracted from both views are integrated and a classifier is trained to predict the correct text characters in the text-CAPTCHA. Figure 2 depicts the architecture of the system.

3.2 Multiview representation

This phase is crucial in the proposed system for two reasons. Firstly, there is no available Multiview dataset for text-CAPTCHA. The datasets available contain single-view data. Therefore, in this phase, data are represented as Multiview data to adapt to the Multiview learning settings. Achieving this was with the help of pre-processing techniques to generate a variant view of each sample from the

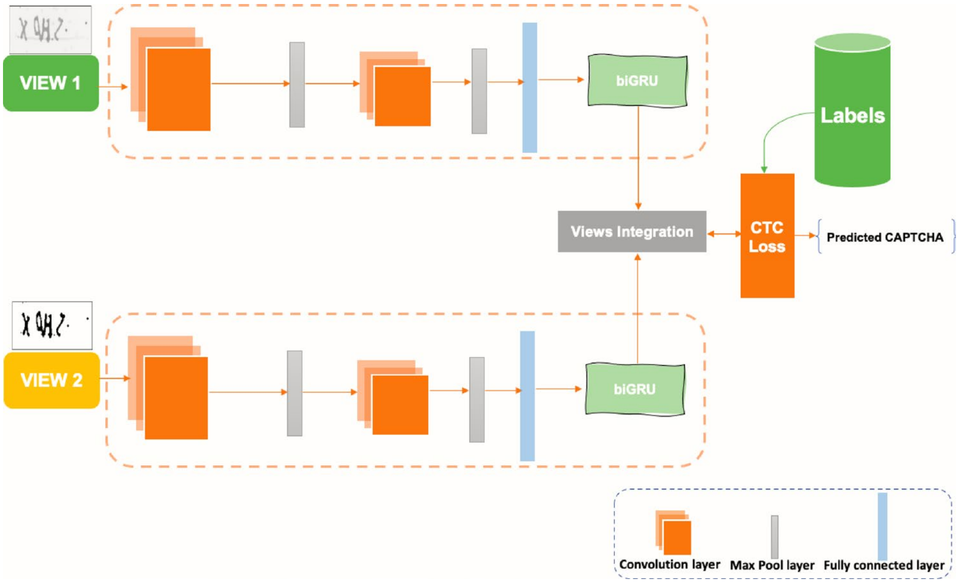


Fig. 2 The proposed Multiview deep learning architecture. Two convolutional blocks are trained to extract spatial features from the Multiview text-CAPTCHA images. The extracted features are further transformed into sequential features using a bi-directional GRU. The extracted features are integrated and a connectionist temporal classification (CTC) loss function is applied

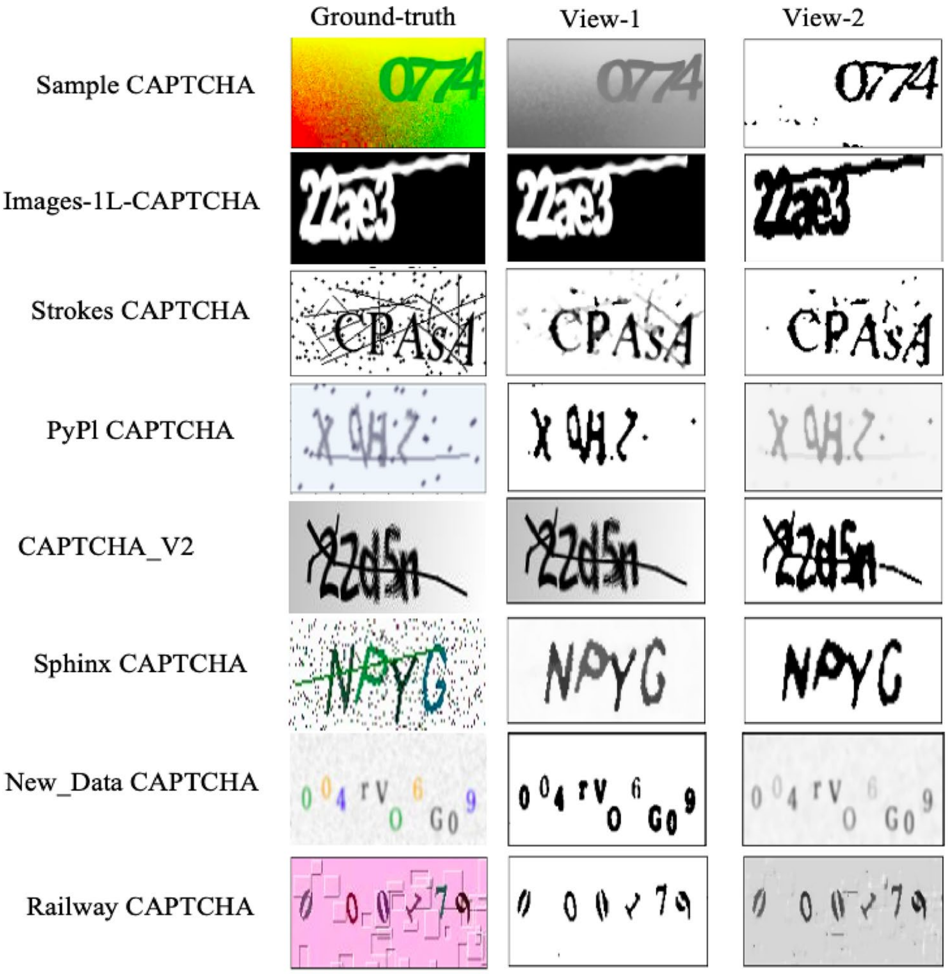


Fig. 3 Randomly selected samples and corresponding generated Multiview data as used in this paper

dataset. Techniques used were; bilateral filtering, thresholding, and median filtering. Choosing these techniques was based on their simplistic implementation and enhancement characteristics. Hence, Multiview data were generated to fit the Multiview setting. Figure 3 shows random examples of the Multiview generated data. Secondly, different kinds of text-CAPTCHA incorporate several security features such as random noise, warping, rotation, coloration, and more. Such features make the CAPTCHA schemes more challenging for computer algorithms to identify. Additionally, the generated Multiview data will help minimize some of the security features embedded in these text-CAPTCHA.

3.3 Feature extraction

In every deep and machine learning model, feature extraction is a critical part that determines the quality of learning of the model. Two kinds of features are extracted in the proposed system, viz; spatial and sequential features. This stage describes the spatial feature extraction. Like many deep learning models, CNN is used to extract spatial features from each view in the proposed system. CNN has shown high success in classification tasks over time [37]. The techniques used in the Multiview representation stage enhance the edges of the text-CAPTCHA image. Therefore, the proposed system uses only a few convolution layers. According to [2], the early layers in a CNN, which are closer to the input, are responsible for learning the edges of the input. With this finding, the proposed system implements two layers of convolutional blocks with rectified linear unit (ReLU) activations, followed by a MaxPooling to downsample the dimension size for the next layer. Each convolution block used a kernel size of 3 X 3 and a stride of 1 X 1 with a padding of 1 X 1. The output of the second layer after MaxPooling is flattened to form a linear layer and passed as input to the next phase. A 10% dropout is applied to prevent over-fitting.

3.4 CAPTCHA character sequencing

This phase is responsible for extracting sequential features from spatial features. Recurrent Neural Network (RNN) [17] has shown impressing results in terms of sequencing continuous information, and hence, considered in the proposed system. Another reason for it's consideration is that the length of characters in some text-CAPTCHA schemes varies, like in the earlier Google text-CAPTCHA. RNN has a sequence to sequence architecture that takes sequential data as input and produces sequential data as output. Unlike CNN, where both input and output are fixed-sized vectors. Therefore, for all input text-CAPTCHA images with variable-length characters, RNN provides the flexibility to determine the length of individual samples during training.

Unfortunately, the vanilla RNN architectures suffer from gradient vanishing or explosion [20]. During the training of a network with backpropagation, if the derivatives of the network's weights are very small, it leads to the phenomenon referred to as gradient vanishing. RNNs with many number of layers are susceptible to this phenomenon. Gradient vanishing causes some neurons to die, such that their gradient is zero. Similarly, when the derivatives of a neuron in the network become very large, the gradient explodes and overshoots the learning curve of the neurons, causing the entire network to be unstable or fail to converge. Backpropagation in RNN is done at each timestep, therefore, increasing the effects of the gradient propagation. Vanilla RNN requires an increase in the number of layers to capture information far away from the current timestamp. An increase in the number of layers increases the number of neurons, which leads to gradient vanishing. Sepp Hochreiter et al. [16] propose to address this problem by introducing a feed-backward neural network. They introduce an input gate, forget gate, and output gate. These gates help each network layer choose what information from the previous timestep to keep, discard, update, and pass as output to the next layer. Figure 4 shows a single unit structure of their proposed LSTM.

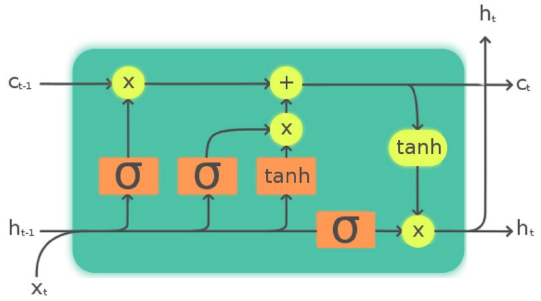


Fig. 4 A structure of a single unit LSTM cell

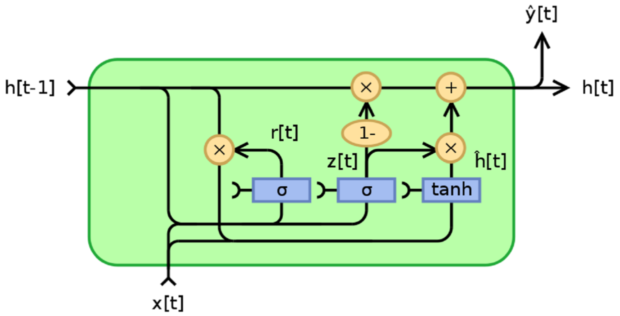


Fig. 5 A structure of a single GRU cell

In 2014, Kyunghyun Cho et al. [8] proposed a minimized LSTM architecture with only two gates (reset and update) instead of three gates as in LSTM. They referred to it as the gated recurrent unit (GRU). The architecture is fast and provides modest accuracy as LSTM. In existing text-CAPTCHA schemes, the average character length ranges from 4 to 10 characters. With this less number of characters, the timesteps required are also less. Hence, the

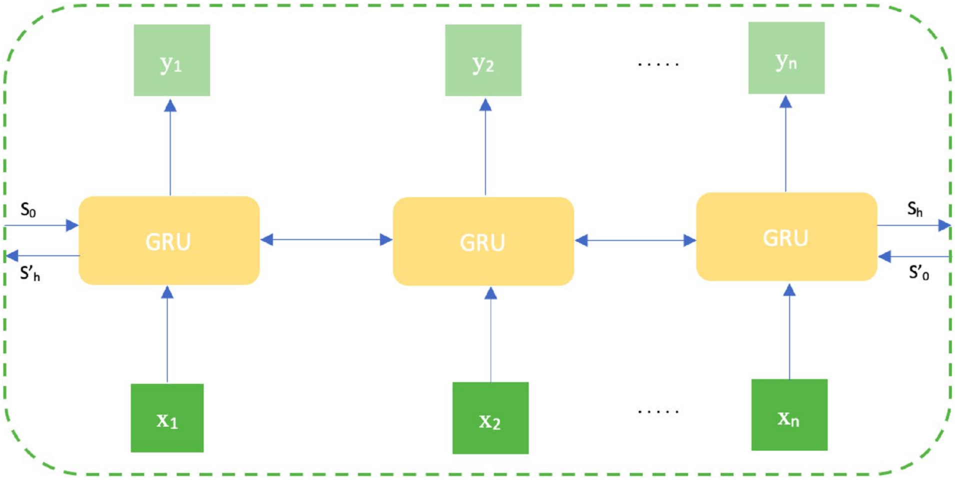


Fig. 6 Showing the architecture of a bi-directional GRU

proposed system implements a GRU to minimize computational resources. The system implements a bidirectional-GRU with two layers and 25% of dropout. Figure 5 depicts a single unit of GRU, and Fig. 6 provides a pictorial description of the bidirectional-GRU architecture as implemented in the proposed system. A detailed expression of a single unit GRU can be seen in equation (1) and (2).

$$\begin{aligned} r[t] &= \sigma(W_{ir}x[t] + b_{ir} + W_{hr}h_{(t-1)} + b_{hr}) \\ z[t] &= \sigma(W_{iz}x[t] + b_{iz} + W_{hz}h[t-1] + b_{hz}) \\ n[t] &= \tanh(W_{in}x[t] + b_{in} + r_t \circ (W_{hn}h[t-1] + b_{hn})) \\ h[t] &= (1 - z_t) \circ n[t] + z[t] \circ h[t-1] \end{aligned} \tag{1}$$

$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{2}$$

where $h[t]$ is the hidden state at timestep t. $h[t-1]$ is the hidden state at time step $t-1$. $x[t]$ is the input at timestep t. $r[t]$ is the reset gate. $z[t]$ is the update gate. $n[t]$ is the new gate. σ is the sigmoid activation function defined in equation (2). *tanh* is the hyperbolic tangent function. W is weight matrices. And lastly $\circ$ is referred to as the Hadamard product.

3.5 View integration and classification/prediction

The final phase of the proposed system begins with Multiview integration. A latent representation is created by stacking the extracted Multiview features together. The latent space ensures that discriminative views complement each other during prediction. This prediction is made on the fly, as the sequential features directly translates into textual predictions. Alex Graves et al. [12] proposed the novel CTC architecture that directly aligns the input sequence to the output sequence. At every timestep and for each sequenced feature, CTC uses a SoftMax activation function to produce a probability distribution of the expected character. Equations 3 and 4 give some technical details of the CTC.

$$y_i^t = \delta(x_t^i) = \frac{e^{x_i^t}}{\sum_0^{\hat{i}} e^{x_j^t}} \tag{3}$$

where δ symbol represent the SoftMax activation function. $\hat{x}_i^t$ represents the *ith* input sequence at timestep t. e is the base of natural logarithm. j is the total number of possible output. And y_i^t is the predicted character of the *ith* sequence at time t. The final CTC loss function is defined in equation (4).

$$\begin{aligned} L(\vec{\mathbf{X}}) &= -\ln \prod_{x_i, y_i \in \vec{\mathbf{X}}} p(y_i \mid x_i) \\ L(\vec{\mathbf{X}}) &= -\sum_{x_i, y_i \in \vec{\mathbf{X}}} \ln p(y_i \mid x_i) \end{aligned} \tag{4}$$

where $p(y_i \mid x_i)$ is the probability of the *ith* output (y_i) given the *ith* input x_i. $\prod$ is the product notation and $\sum$ is the sum notation. ln represents the base e logarithm function. $\vec{\mathbf{X}}$ is the set of given data. A maximum likelihood estimation is employed to optimize this loss function.

4 Experimental setup

Experiments on the proposed system were conducted in a Python 3.7.6 environment, on a personal computer with 1.7GHz (Dual-core), Intel Core-i7 processor, 8GB (DDR3) RAM, running on macOS Catalina 10.15.3.

4.1 Datasets

For the sake of security, text-CAPTCHA datasets are not publicly made available. Therefore, text-CAPTCHA datasets are synthesized using the security features implemented in SOTA schemes. However, some already synthesized datasets used in other studies were made public and were used to evaluate the proposed system. This paper used a total of eight (8) different datasets. In each dataset, training data takes up 90%, and testing takes up 10%. Table 1 itemizes all the datasets used along with their security features and the number of text-CAPTCHA images contained in each dataset. Figure 7 displays a text-CAPTCHA sample from each dataset.

4.2 Dataset pre-processing

The pre-processing technique implemented enhances the edges of the text-CAPTCHA image. These pre-processing techniques vary in each dataset. Table 2 summarized the pre-processing techniques used in each dataset.

Some basic pre-processing techniques were common in all datasets, such as;

i *Grayscale conversion:* Text-CAPTCHA images were converted to Grayscale from RGB to ensure color uniformity and to reduce the number of parameters to be learned by the proposed networks.

ii *Image resizing:* Characters in text-CAPTCHA images are usually written horizontally. Therefore, the width of text-CAPTCHA images is more crucial than its height. Text-CAPTCHA images in the proposed system are downsized by 25% and 10% on height and width, respectively.

Some other pre-processing techniques were implemented specifically for some certain dataset. These techniques are described below;

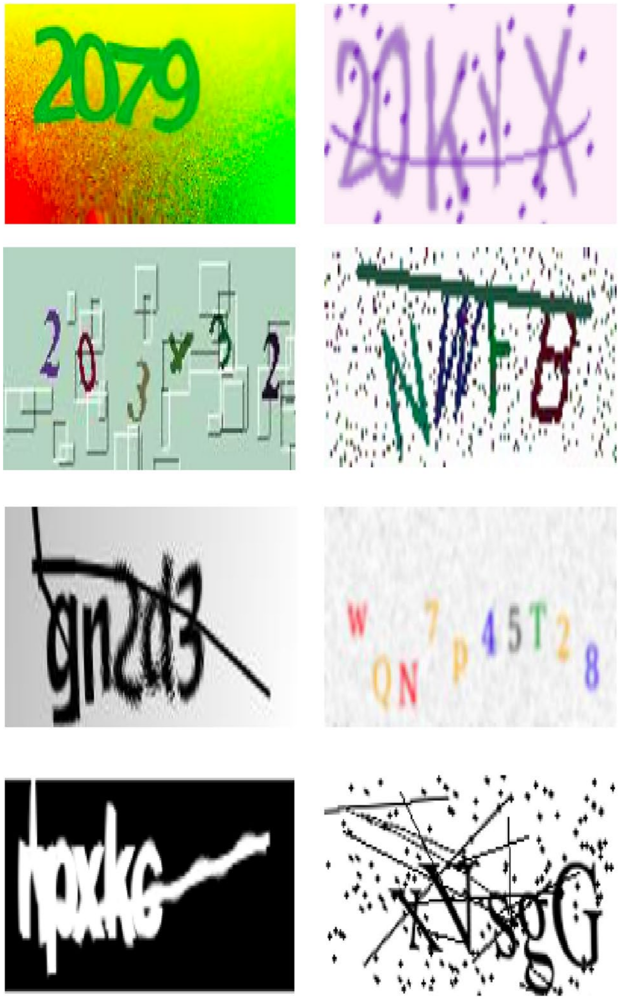


Fig. 7 Random samples of the dataset used

i *Median filtering* [3]: This technique highlights the edges of the image and reduces the background noise in the image by introducing blurry effects from the neighborhood analysis. It plays a significant role in increasing the accuracy of the proposed system and decreasing the

Table 1 Showing the CAPTCHA schemes used in this study with their security features

Dataset	Rotation	Warping	Background noise	Random lines/arcs	Random coloration	Overlapping	Size (in number of data sample)
CAPTCHA_V2	✓	✓	–	✓	–	✓	2140
Pypl CAPTCHA	✓	–	✓	✓	–	✓	604800
Railway CAPTCHA	✓	–	✓	–	✓	–	100K
Sphinx CAPTCHA	✓	–	✓	✓	✓	✓	990K
Images-1L-CAPTCHA	✓	✓	–	✓	–	✓	100K
Strokes CAPTCHA	✓	–	✓	✓	–	✓	10K
Sample CAPTCHA	–	✓	✓	–	✓	✓	25K
New_Data CAPTCHA	–	–	✓	–	✓	–	10K

Table 2 The pre-processing techniques implemented in the CAPTCHA dataset used in this study

Dataset	Median filtering	Threshold	Bilateral filtering
CAPTCHA_V2	✓	✓	–
Images-1L-CAPTCHA	–	✓	✓
PyPl CAPTCHA	✓	✓	–
Railway CAPTCHA	✓	✓	–
Sphinx CAPTCHA	✓	✓	–
Strokes CAPTCHA	✓	✓	–
Sample CAPTCHA	✓	✓	✓
New_Data CAPTCHA	–	✓	✓

Table 3 Showing the average time taken to break the text-CAPTCHA schemes used in this study

Dataset	Time taken (in seconds)
CAPTCHA_V2	0.0032
Images-1L-CAPTCHA	0.134
PyPl CAPTCHA	0.21
Railway CAPTCHA	0.067
Sphinx CAPTCHA	0.0514
Strokes CAPTCHA	0.0832
Sample CAPTCHA	0.1129
New_Data CAPTCHA	0.132

training time of the proposed system. In the proposed system, a nested optimized median filter [3] was implemented to achieve better results.

ii *Threshold:* This technique performs lightweight image segmentation of the background and foreground in the text-CAPTCHA image. The filter minimizes the task of the CNN and improves its training time. Binary inverse and the OTSU threshold algorithm [26] were combined to achieve this task.

iii *Bilateral filtering*[32]: This technique is a non-linear and non-iterative filtering process. It computes the averages of non-edge pixels of a given image. Therefore, highlighting the edges of the image.

4.3 Performance evaluation metrics

A CAPTCHA scheme must demonstrate that it is protective against any computer bots, such that, given a CAPTCHA test, no computer bot should achieve above 1% of accuracy in solving the CAPTCHA. Therefore, to evaluate the effectiveness of the proposed system. The system must achieve an accuracy above 1% in breaking the CAPTCHA.

The accuracy of the proposed system was calculated using the formula below;

$$Accuracy = \left(\frac{TP + TN}{TP + TN + FP + FN}\right) * 100 \tag{5}$$

where *TP* is the number of correct text-CAPTCHA images the system correctly predicts. *TN* is the number of incorrect text-CAPTCHA the system incorrectly predicts. *FP* is the number of incorrect text-CAPTCHA the system correctly predicts. And *FN* is the number of incorrect text-CAPTCHA the system incorrectly predicts.

Another essential metric needed to evaluate the proposed system is the time taken to break each text-CAPTCHA. Table 3 reports the average time taken to compute and predict the CAPTCHA in each of the datasets used in this study.

4.4 Proposed system evaluation with existing systems

This section describes the comparison of the proposed system's performance with existing state-of-the-art systems. [41] recently conducted an experimental study on breaking text-CAPTCHA images across 32 different text-CAPTCHA schemes. The schemes are from the top 50 websites ranked by Alexa. However, most of the CAPTCHA scheme datasets are not publicly made available for security and privacy reasons. Meanwhile, several security features constitute the make-up of text-CAPTCHA schemes. Technically, defeating different text-CAPTCHA schemes with similar security features is possible with a unique text-CAPTCHA breaking system. The evaluation of the proposed system performance is compared to that of [19, 41]. And where datasets are not publicly available, the comparison is made with schemes that share corresponding security features. Whereas, where datasets are publicly available, evaluation is done on the datasets and compared with existing systems. Table 4 demonstrates the similarities with regard to the corresponding security features of text-CAPTCHA datasets as used in this paper.

5 Results and discussion

In this section, the performance of the proposed system is analyzed, reported, and discussed. The evaluation process of the proposed system is conducted in three stages.

Table 4 Dataset security features in the proposed system compared to dataset security features in existing systems. [1] refers to datasets used in [41]. [2] refers to datasets used in [19]

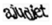
Proposed system	Example	Existing systems	Example	Security features
Railway CAPTCHA		Dig	[1]	Background noise, Rotation, Random coloration.
Images-1L-CAPTCHA	[1]	Baidu(2013)	[1]	Rotation, Warping, Random arcs, Overlapping.
Strokes CAPTCHA		Slashdot	[1]	Rotation, Background noise, Random arcs, Overlapping.
PyPl CAPTCHA		Sina	[1, 2]	Rotation, Background noise, Random arcs, Overlapping
Sample CAPTCHA		Taobao, reCAPTCHA (2011) and Tencent	[1] [1] [2]	Warping, Background noise, Overlapping

Table 5 Showing the accuracy result for the text-CAPTCHA schemes used in this study

Dataset	Accuracy	Epoch	Training time
CAPTCHA_V2	100%	5	16mins 35s
Images-1L-CAPTCHA	100%	6	35mins 16s
Pypl CAPTCHA	94%	25	1hr 23mins
Railway CAPTCHA	98%	25	1hr 08mins
Sphinx CAPTCHA	97.3%	25	1hr 52mins
Strokes CAPTCHA	97%	25	1hrs 26mins
Sample CAPTCHA	97.8%	38	1hr 9mins
New_Data CAPTCHA	93.6%	26	1hr 22mins

5.1 Quantitative evaluation analysis of the proposed system

The quantitative evaluation of the proposed system is analyzed and discussed in this section. The quantitative evaluation in this case refers to the quantitive performance of the proposed system with reference to two important metrics; accuracy and execution time. Accuracy is one of the important metrics for evaluating the effectiveness of CAPTCHA breaking systems. For an effective CAPTCHA scheme, the accuracy of any computer-based system must be less than 1%. Above that is considered unsafe for users. As reported in Table 5, the proposed system's accuracy is at least 93.6% in all the datasets that were used in this paper. Therefore, proving that the proposed system is highly effective in breaking the text-CAPTCHA schemes. Furthermore, the execution time or time taken to break a CAPTCHA is another essential metric used in evaluating the effectiveness of the proposed system. Table 3 reported the average time taken to break the text-CAPTCHA in each dataset. The PyPl CAPTCHA scheme requires the maximum time (0.21 seconds) to break due to the CAPTCHA scheme's high complexity, which requires more time to process. This is a tremendous amount of time to achieve for breaking any CAPTCHA scheme. Hence, it is safe to conclude that the proposed system effectively breaks the text-CAPTCHA schemes used in this paper with excellent speed and accuracy.

5.2 Ablation study

While trying to secure text-CAPTCHA schemes, existing systems tried several combinations of complex security features to confuse and make text-CAPTCHA schemes more challenging to solve for computer bots. Meanwhile, such kind of approach may prove troublesome for computer bots. However, it could become difficult for humans to solve too. Therefore, negating the fundamental principle that characterizes an effective CAPTCHA scheme. For this reason, this study deemed it crucial to evaluate the effectiveness of the text-CAPTCHA schemes used in this paper on human users. This paper conducts an ablation study to demonstrate the performance of human users on the text-CAPTCHA schemes used in this paper. The results obtained are compared to the performance of the proposed system. Further analysis of the ablation study demonstrates the effect of security features in text-CAPTCHA images on human users. The study covers a total number of 71 students and non-student. Student participants were from different universities, polytechnics, colleges, and high schools. The majority of the participant resides in India. Because CAPTCHA schemes were designed not to discriminate among age groups, all age groups were considered and allowed to participate in the study. The ablation study was conducted online due to the COVID-19 pandemic. The study comprises eight sections, and each of the sections covers a specific kind of text-CAPTCHA scheme used in this paper. In each section, five samples of text-CAPTCHA images were displayed to the user and asked the participants to type in the corresponding text characters. Each participant participated of their own volition, and did not receive any remunerations throughout. Figure 8 depicts

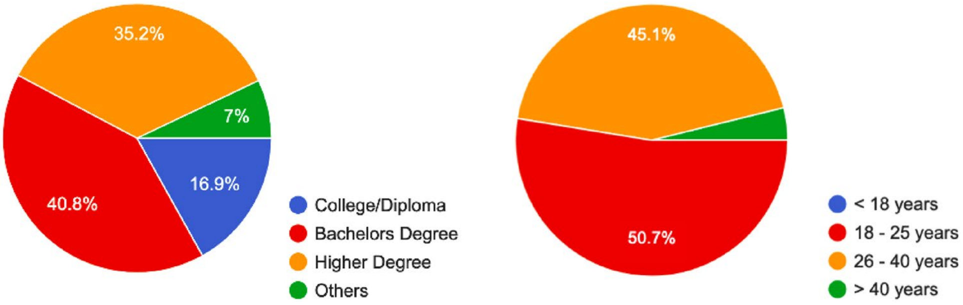


Fig. 8 Piechart showing the demography of participant that participates in the ablation study

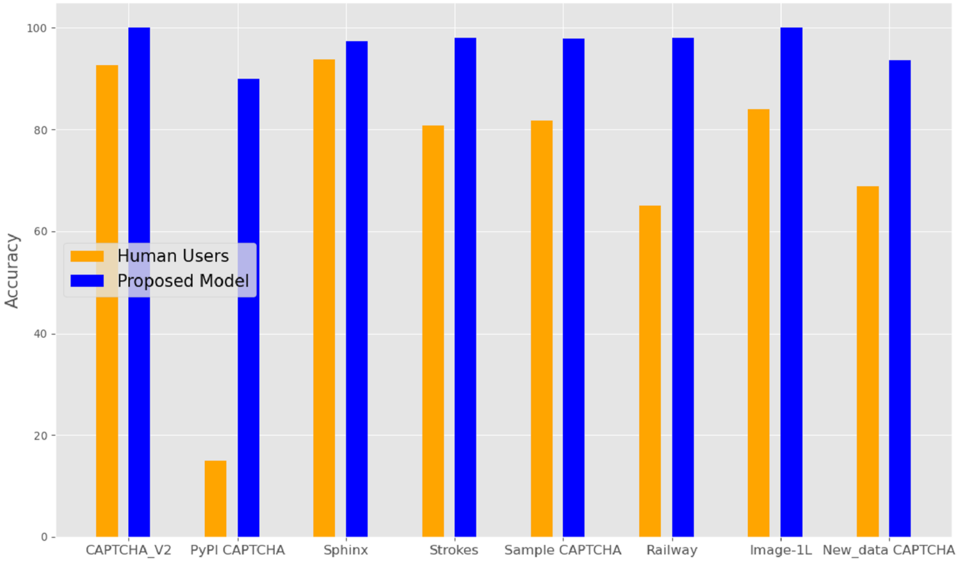


Fig. 9 Chart comparing the performance accuracy of human users and the proposed system on all the text-CAPTCHA dataset used in this paper

a piechart of the demography of the participants in the study. Students currently pursuing or has completed a bachelor's degree program are the majority of the participant. And the majority of age groups are participants within the range of 18-25 years then 26-40 years old. Figure 9 displays the chart results for the performance of human users in the ablation study against the performance of the proposed system in terms of accuracy. The figure demonstrates that where there is a drop or rise in accuracy for human users, the same effect reflects in the proposed system. Another insight to note from the chart is that the accuracy of human users is lower in text-CAPTCHA schemes with more complex security features. This performance deterioration reflects in the performance of the proposed system too. Hence, the scheme's effectiveness reduces as the security features of the text-CAPTCHA are increased. Overall, the performance of the proposed system outperforms the human users in all the text-CAPTCHA datasets used in this study (Fig. 10).

5.3 Performance comparison with existing systems

The quantitative evaluation of the proposed system was carried out in two phases. The first phase evaluates and compares the performance of the proposed system to the existing studies whose dataset is available publicly. As reported in Table 6, the proposed system has achieved an accuracy of almost 40% higher than existing systems. The second phase evaluates and compares the performance of the proposed system to the existing studies whose datasets are not available publicly. In this phase, a relative comparison to corresponding text-CAPTCHA schemes in terms of security features. Table 7 report this comparison. The report shows that the proposed system outperforms the existing state-of-the-art systems.

6 Conclusion and future work

A Multiview deep learning paradigm for breaking text-CAPTCHA schemes was introduced. The Multiview learning setting aims to explore deep complementary features from multiple views of the input text-CAPTCHA image. The proposed architecture first utilizes a combination of CNN (for spatial feature extraction) and a bidirectional GRU (to learn sequential features). Then a view integration technique that maximizes the complementarity between

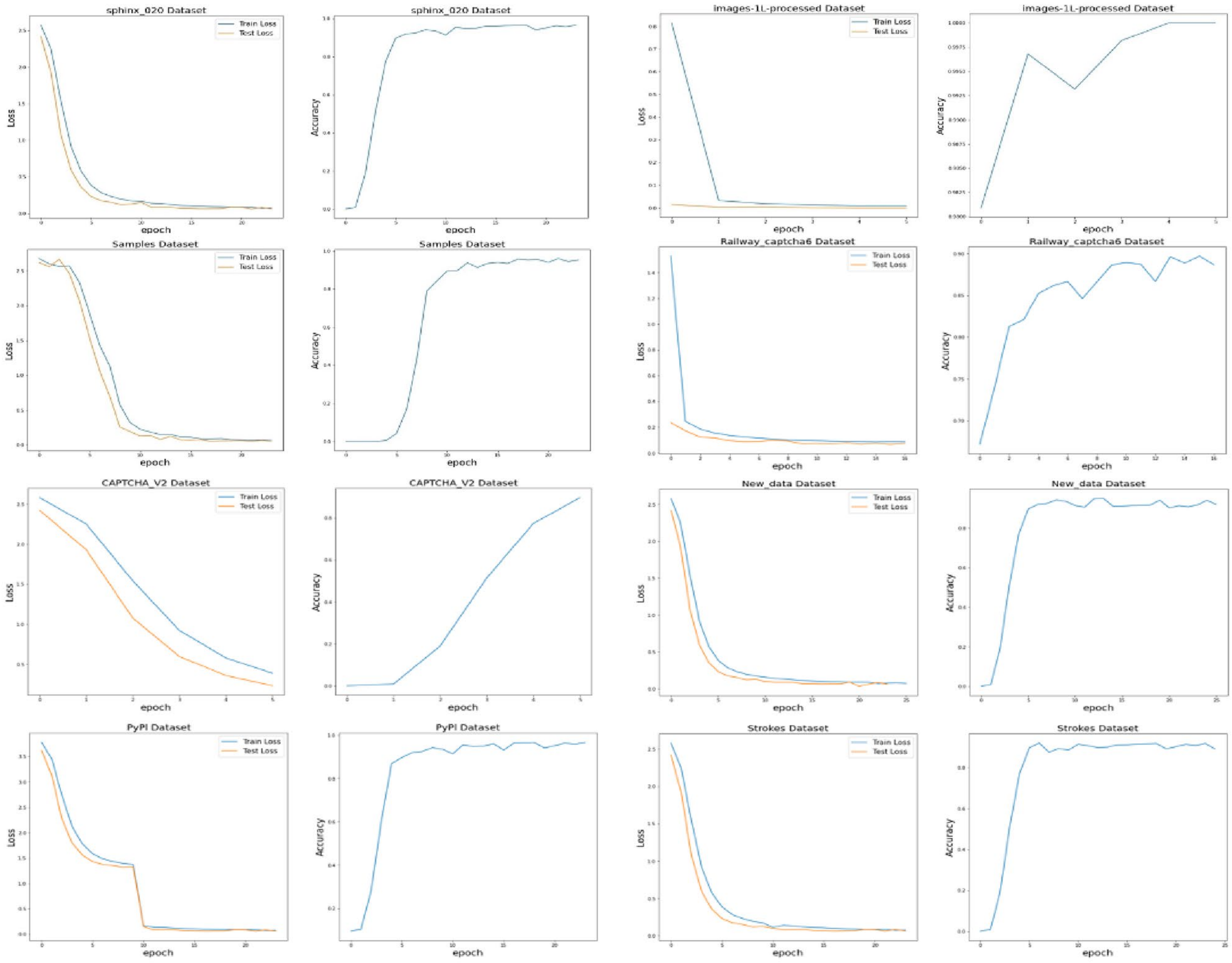


Fig. 10 Graphs demonstrating the train loss, test loss, and accuracy of the proposed system during training

Table 6 Comparing the performance of the proposed system with existing systems on the CAPTCHA_V2 dataset

CAPTCHA Schemes	Accuracy
Elie et al. [5]	51.1%
Ye et al. [41]	51.6%
Proposed system	**100%**

both discriminative views is employed. Lastly, the integrated views are mapped into expected output predictions and trained end-to-end using a CTC loss function. The proposed system was trained and validated across eight different datasets. The system achieves remarkably high accuracies within the ranges of 93.6% to 100% across the eight datasets. Impressively, the system requires very little time to break the text-CAPTCHA with an average time between 0.0032% to

Table 7 Proposed system performance compared to corresponding text-CAPTCHA schemes in existing systems with respect to their security features

A: Proposed system	Accuracy (A)	B: Existing system [41]	Accuracy (B)	C: Existing system [19]	Accuracy (C)
Railway CAPTCHA	98%	Dig	95%	–	–
Images-1L-CAPTCHA	100%	Baidu(2013)	89%	–	–
Strokes CAPTCHA	97%	Slashdot	86.4%	–	–
PyPl CAPTCHA	94%	Sina	90%	Sina	90%
Sample CAPTCHA	97.8%	Taobao, reCAPTCHA (2011)	90.7%, 87.4%	Tencent	75.4%

0.21% seconds. To further demonstrate the performance of the proposed system, an ablation study was conducted on a diverse group of human users. The study shows that, during the design of text-CAPTCHA schemes, when the complexity of difficulty levels is high, the text-CAPTCHA scheme is not very efficient for human users. A comparison of the ablation study results with the proposed system demonstrated that the system outperforms the human users that the CAPTCHA is designed to serve. Further comparison shows that the proposed system outperformed existing state-of-the-art text-CAPTCHA breaking systems. Future works will focus on optimizing the decoder of the CTC architecture to increase training time. Proposing a new CAPTCHA scheme that can resist the existing CAPTCHA solutions.

References

1. Alzubi JA, Jain R, Nagrath P et al (2021) Deep image captioning using an ensemble of cnn and lstm based deep neural networks. J Intell Fuzzy Syst 40(4):5761–5769
2. Andrearczyk V, Whelan PF (2017) Chapter 4 - deep learning in texture analysis and its application to tissue image classification. In: Depeursinge A, Al-Kadi O, Mitchell J (eds) Biomedical Texture Analysis. The Elsevier and MICCAI Society Book Series, Academic Press, p 95–129, https://doi.org/10.1016/B978-0-12-812133-7.00004-1, https://www.sciencedirect.com/science/article/pii/B9780128121337000041
3. Appiah O, Asante M, Hayfron-Acquah JB (2020) Improved approximated median filter algorithm for real-time computer vision applications. J King Saud Univ Comput Inf Sci
4. Baird HS, Popat K (2002) Human interactive proofs and document image analysis. In: International Workshop on Document Analysis Systems, Springer, pp 507–518
5. Bursztein E, Moscicki A, Fabry C, et al (2014) Easy does it: More usable captchas. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp 2637–2646
6. Chellapilla K, Larson K, Simard PY, et al (2005) Building segmentation based human-friendly human interaction proofs (hips). In: International Workshop on Human Interactive Proofs, Springer, pp 1–26
7. Chen J, Luo X, Liu Y et al (2019) Selective learning confusion class for text-based captcha recognition. IEEE Access 7:22246–22259
8. Chung J, Gulcehre C, Cho K, et al (2014) Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv preprint arXiv:1412.3555
9. Gao H, Tang M, Liu Y et al (2017) Research on the security of microsoft's two-layer captcha. IEEE Trans Inf Forensics Secur 12(7):1671–1685
10. Gönen M, Alpaydın E (2011) Multiple kernel learning algorithms. J Mach Learn Res 12:2211–2268
11. Gönen M, Khan S, Kaski S (2013) Kernelized bayesian matrix factorization. In: International Conference on Machine Learning, PMLR, pp 864–872
12. Graves A, Fernández S, Gomez F, et al (2006) Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In: Proceedings of the 23rd international conference on Machine learning, pp 369–376
13. Guan Y, Wei Q, Chen G (2019) Deep learning based personalized recommendation with multi-view information integration. Decis Support Syst 118:58–69
14. Guerar M, Verderame L, Migliardi M, et al (2021) Gotta captcha'em all: A survey of twenty years of the human-or-computer dilemma. arXiv preprint arXiv:2103.01748
15. He C, Peng L, Le Y, et al (2019) Dynamically weighted multi-view semi-supervised learning for captcha. In: Pacific-Asia Conference on Knowledge Discovery and Data Mining, Springer, pp 343–354
16. Hochreiter S, Schmidhuber J (1997) Long short-term memory. Neural Comput 9(8):1735–1780
17. Jordan MI (1997) Serial order: A parallel distributed processing approach. Advances in psychology, vol 121. Elsevier, Amsterdam, pp 471–495
18. Kim S, Choi S (2019) Dotcha: A 3d text-based scatter-type captcha. In: International Conference on Web Engineering, Springer, pp 238–252
19. Li C, Chen X, Wang H et al (2021) End-to-end attack on text-based captchas based on cycle-consistent generative adversarial network. Neurocomputing 433:223–236
20. Li S, Li W, Cook C, et al (2018) Independently recurrent neural network (indrnn): Building a longer and deeper rnn. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 5457–5466
21. Liu X, Zhu X, Li M et al (2018) Late fusion incomplete multi-view clustering. IEEE Trans Pattern Anal Mach Intell 41(10):2410–2423
22. Liu X, Zhu X, Li M et al (2019) Multiple kernel k k-means with incomplete kernels. IEEE Trans Pattern Anal Mach Intell 42(5):1191–1204
23. Ma Y, Zhong G, Liu W et al (2020) Neural captcha networks. Appl Soft Comput 97(106):769 https://doi.org/10.1016/j.asoc.2020.106769, www.sciencedirect.com/science/article/pii/S1568494620307079
24. Nachar RA, Inaty E, Bonnin PJ et al (2015) Breaking down captcha using edge corners and fuzzy logic segmentation/recognition technique. Sec Commun Netw 8(18):3995–4012
25. Nouri Z, Rezaei M (2020) Deep-captcha: a deep learning based captcha solver for vulnerability assessment. Available at SSRN 3633354
26. Otsu N (1979) A threshold selection method from gray-level histograms. IEEE Trans Syst Man Cybern 9(1):62–66
27. Ouyang Z, Zhai X, Wu J et al (2021) A cloud endpoint coordinating captcha based on multi-view stacking ensemble. Comput Secur 103(102):178
28. Pratomo AH, Nugraha AF, Siswantoro J, et al (2019) Algorithm border tracing vs scanline in blob detection for robot soccer vision system. International Journal of Advances in Soft Computing & Its Applications 11(3)
29. Rui C, Jing Y, Rong-gui H et al (2013) A novel lstm-rnn decoding algorithm in captcha recognition. 2013 Third International Conference on Instrumentation. Measurement, Computer, Communication and Control, IEEE, pp 766–771
30. Shao R, Shi Z, Yi J, et al (2021) Robust text captchas using adversarial examples. arXiv preprint arXiv:2101.02483
31. Tao Z, Liu H, Li S, et al (2017) From ensemble clustering to multi-view clustering. In: IJCAI
32. Tomasi C, Manduchi R (1998) Bilateral filtering for gray and color images. In: Sixth international conference on computer vision (IEEE Cat. No. 98CH36271), IEEE, pp 839–846
33. Turing A (1950) Computing machinery and intelligence. Perspectives on the computer revolution
34. Vert JP (2003) Kernel methods in computational biology. Kyoto Univ Res Inf Repos 81(1):142–155

35. Von Ahn L, Blum M, Hopper NJ, et al (2003) Captcha: Using hard ai problems for security. In: International conference on the theory and applications of cryptographic techniques, Springer, pp 294–311
36. Wang P, Gao H, Shi Z et al (2020) Simple and easy: transfer learning-based attacks to text captcha. IEEE Access 8:59044–59058
37. Wang P, Fan E, Wang P (2021) Comparative analysis of image classification algorithms based on traditional machine learning and deep learning. Pattern Recogn Lett 141:61–67
38. Wang Y, Wei Y, Zhang M et al (2021) Make complex captchas simple: a fast text captcha solver based on a small number of samples. Inf Sci 578:181–194
39. Xie X, Sun S (2013) Multi-view clustering ensembles. In: 2013 International Conference on Machine Learning and Cybernetics, IEEE, pp 51–56
40. Xu C, Tao D, Xu C (2013) A survey on multi-view learning. arXiv preprint arXiv:1304.5634
41. Ye G, Tang Z, Fang D et al (2020) Using generative adversarial networks to break and protect text captchas. ACM Trans Privacy Secur (TOPS) 23(2):1–29
42. Yu H, Xiao S, Yu Z et al (2019) Imcaptcha: imperceptible captcha based on cursor trajectories. IEEE Consum Electron Mag 9(1):74–82
43. Zhang N, Ebrahimi M, Li W, et al (2020) 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), IEEE, pp 1–6
44. Zhang N, Ebrahimi M, Li W et al (2022) Counteracting dark web text-based captcha with generative adversarial learning for proactive cyber threat intelligence. ACM Trans Manag Inf Syst. https://doi.org/10.1145/3505226
45. Zi Y, Gao H, Cheng Z et al (2019) An end-to-end attack on text captchas. IEEE Trans Inf Forensics Secur 15:753–766

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.