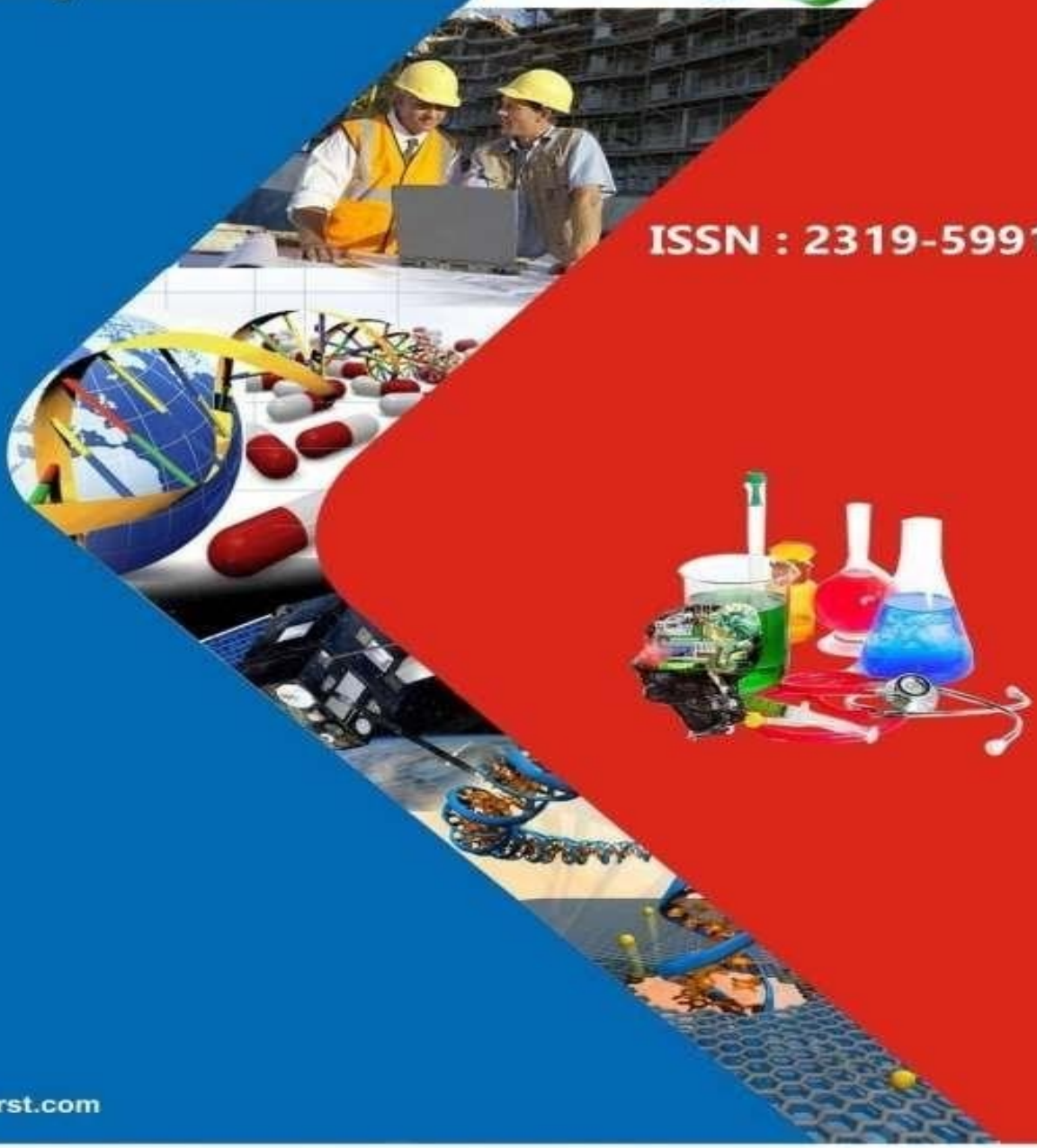
International Journal of Engineering Research and Science & Technology

IJERST

ISSN : 2319-5991



www.ijerst.com

Email: editor@ijerst.com or editor.ijerst@gmail.com

International Journal of Engineering Research and Science & Technology

ISSN 2319-5991 www.ijerst.com

Vol. 21, Issue 2, 2025

The Capsecure Project: Analyzing and Recognizing Captchas using a Custom-Based CNN Model

[1]V.B.N. AJAY, [2]D GOPIREDDY

[1,2] Assistant Professor, Department of Computer Science and Engineering,

NVR COLLEGE OF ENGINEERING & TECHNOLOGY -TENALI

Mail: vbnajaycse@gmail.com

Abstract—

To prevent attacks from programs or other computerized agents that try to mimic human intellect, CAPTCHAs are automated tests that try to discriminate between computers and people. The overarching goal of this study is to create a CAP-SECURE, a custom-based convolutional neural network (CNN) model, that can bypass CAPTCHA. Differentiating or informing websites about the CAPTCHAs' shortcomings and vulnerabilities is the goal of the suggested approach. Compared to other CNN architectures, such as VGG-16 and ALEX-net, the CAP-SECURE model—which is based on a sequential CNN model—performs better. In theory, the model could decipher and investigate both numerical and alphabetical CAPTCHAs. In order to train our model effectively, we have created a dataset consisting of 2,00000 CAPTCHAs. To address the present problems and give solutions for CAPTCHAs, we examine a CNN-based deep neural network model in this presentation. For the alpha-numerical test dataset, the network's cracking accuracy is shown to be 94.67%. The suggested custom-based model outperforms more conventional deep learning approaches in terms of both recognition rate and resilience.

Keywords - Machine learning, CAP-SECURE, convolutional neural network, and CAPTCHA recognition

I. INTRODUCTION

Completely Automated Public Turing Test to Distinguish Between Computers and Humans, or CAPTCHA, is a system that uses automated testing to differentiate between robots and humans. the human race [2]. Several fields have found extensive usage for it, including data security and network protection. Concerns about network security are only becoming worse as internet technology develops further. In order to protect the privacy of users and their online services from a variety of cyber threats, assaults, and other intrusions, CAPTCHAs are used on the internet [1]. It has recently been adopted by several prominent mainstream websites. To be more precise, in many cases, these assaults result in the replacement of people by computer programs. Computer programs attempt to automate services in order to send a flood of unwanted emails, get access to databases, or manipulate online pools. The integration of CAPTCHA recognition with AI and image processing is crucial to the advancement of AI technology; it has several applications in areas such as optical character recognition, handwriting recognition, license plate recognition, and more. Visual representations such as numerical or alpha-numerical strings, audio, or image sets form the basis of the popular CAPTCHA. At the same time, text-based CAPTCHAs are widely employed in their respective fields. Figure 1 displays many forms of alpha-numerical CAPTCHAs along with some instances of them. Numerous techniques exist for enhancing the complexity of CAPTCHAs. by supplementing them with effective noise and disruptions [20], [4]. For example, superimposing zigzag lines over an alphanumeric CAPTCHA is one way to increase the scheme's security. Figure 2 displays many instances of CAPTCHAs that have been affected by noise distortion. Visual CAPTCHAs are quickly gaining popularity alongside their text-based counterparts. In these CAPTCHAs, the user is asked to identify a certain item from a collection of photos that include things like traffic lights, cars, street signs, landscapes, and sculptures. Figure 3 shows an example of a CAPTCHA that uses images. Picture preprocessing, segmentation of characters, character identification, feature extraction, and post-processing are the main phases in traditional CAPTCHA recognition. Segmentation and feature extraction are crucial steps in

International Journal of Engineering Research and Science & Technology

ISSN 2319-5991 www.ijerst.com

Vol. 21, Issue 2, 2025

these models that determine their accuracy and efficiency. In this piece, we'll take a look at common CAPTCHA graphics, which are built on random English characters and numbers. Additionally, they may be easily generated. Also, another use of CAPTCHA that deserves note is in OCR (Optical Character Recognition). Despite its robustness, current optical character recognition algorithms have several limitations when it comes to identifying various hand-written scripts or weakened handwriting [6]. Convolutional neural networks (CNNs) have recently outperformed more conventional approaches to picture identification, demonstrating its potential as a kind of deep neural networks. Traditional methods of pixel point extraction and template matching are limited to detecting basic CAPTCHAs. One major benefit of CNN over more conventional pattern recognition methods is its ability to actively learn features without the need for artificial design. Bypassing data pretreatment issues, the derived picture characteristics exhibit robust expressive capacity. Despite CNN's success, research on the recognition impact of complicated CAPTCHA is lacking [7]. Then, in order to detect the CAPTCHAs, a CNN algorithm is suggested by [8]. The CAP-SECURE model is the center of attention in this research. It is a CNN based image CAPTCHA recognition system. Picture CAPTCHAs include both English characters and random numbers. Making it is a breeze, and the physical force defenses are tough to overcome. In an effort to fix CAPTCHA recognition and identify its flaws, this study sets out to do just that. So that these websites can withstand intelligence assaults and bots, there has been an attempt to enhance CAPTCHA technology and create more robust CAPTCHAs.

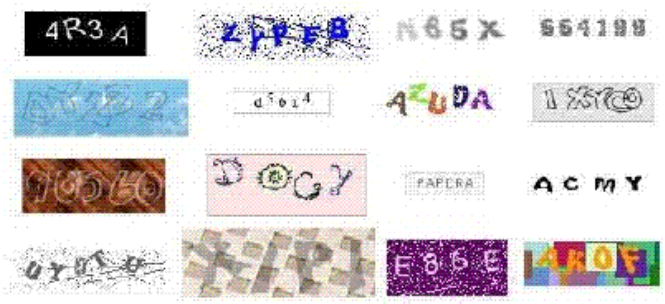


Fig. 1: Examples of various alpha-numeric CAPTCHAs

Fig. 1: Examples of five-digit distorted CAPTCHAs

International Journal of Engineering Research and Science & Technology

ISSN 2319-5991 www.ijerst.com

Vol. 21, Issue 2, 2025

Fig. 2: Image based Captcha

This is the general format of the paper: In Section II, the literature pertinent to CAPTCHA advancements is reviewed. Part III provides an overview of our planned method. Section IV delves into the discussion of the experimental outcomes. Section V concludes with a few last thoughts.

II. BACKGROUND

In this part, we will go over what is known about CAPTCHA and what has happened recently in the area. One popular approach is segmentation recognition, where Recognizing CAPTCHA. The standard procedure for locating a specific area in a picture that contains a number or character, then segmenting the image and identifying the individual characters [9]. Take the authors of [10] as an example; they demonstrated how character segmentation in CAPTCHA may greatly enhance recognition accuracy. By manipulating pixels, the authors of [11] were able to decipher this kind of CAPTCHA. The same author went on to successfully segment it using the projection method, achieving a success rate of 60% in cracking and an accuracy of up to 90% in the process. Then, in [12], the character strokes are extracted using the Gambor filtering approach. They achieved a 77% success rate in CAPTCHA cracking after using the graph search approach to determine the best combination for character segmentation. The use of a shape context technique for CAPTCHA picture identification is covered in [13]. Deep learning-based CAPTCHA identification systems have recently gained traction in the academic community. For text-based CAPTCHA recognition without segmentation, the authors of [14] have suggested a multilabel CNN. Character distortion and complicated CAPTCHA have both been improved upon. A convolutional recurrent neural network was suggested by the authors of [15] to achieve general CAPTCHA recognition by combining CNN with RNN. For superior recognition results with CAPTCHAs of variable length sequences, see the fast region-based CNN for overall recognition in [16]. The authors of [17] trained a convolutional neural network to recognize CAPTCHA's stroke and other unique characteristics. By including distortion, rotation, and background noise, they significantly enhanced the CAPTCHA's identification accuracy. When it comes to classification and recognition, deep neural networks are far more efficient than the old ways because of their superior learning capacity [18]. Several writers have proposed adding noises, such line cross noise or point-based scattering noise, to CAPTCHA designs in order to increase their complexity and security [20] and [21]. From [22] to [25], we may find approaches that are based on CNNs. For improved outcomes, Kwon et al. [26] have used CNN with transfer style. In contrast to DenseNet [28], a CNN with

International Journal of Engineering Research and Science & Technology
ISSN 2319-5991 www.ijerst.com
Vol. 21, Issue 2, 2025

a few tweaks was analyzed in [19]. A deep neural network trained in Python was previously investigated by Garg and Pollett [29] for the purpose of solving fix-lengthening CAPTCHAs. A network with two Convolutional Maxpool layers has been considered. Rather of using simple dense layers, their approach showcases recurrent layers based on Stochastic Gradient Descent (SGD) with Nesterov momentum. Additionally, it has been shown that thick layers provide superior precision. Afterwards, a web-browser-based system for illuminating photo CAPTCHAs was proposed by Sivakorn et al. [5]. To get around this problem, the writers of [25] used a CNN. Three convolutional layers and two thick layers were used. On top of that, they have used a technique to lessen the size of the necessary training dataset. Researchers Lu Wang et al. [30] looked at merging characters recognition and came up with a strategy using minimal projection and local minimum values. Additionally, terms that don't seem plausible may be looked up in a dictionary. To overcome CAPTCHAs, for instance, Mori and Malik [18] suggested a strategy that involves a dictionary containing all 411 words.

III. PROPOSED METHOD

In recent years, deep learning has grown in significance within scientific study. It has been quite successful in several domains, including voice recognition, picture language processing, object identification, and recognition. The ability to actively learn characteristics without the need for artificial design is deep learning's primary benefit. Use of CAPTCHAs on heavily trafficked websites makes them very susceptible to assaults by bots that attempt to decipher them without human intervention. assistance from a human being. We propose CAP-SECURE, a deep neural network, to detect CAPTCHAs based on the aforementioned findings and ideas. By tailoring convolutional layers to meet our specifications, we build a specialized deep neural network that we call CAP-SECURE. Presented here is the comprehensive approach of managing, recognizing, and dividing the alphanumerical Captcha images. The suggested approach takes an image as its only input, necessitating picture pre-processing, output encoding, and network architecture. Part One: Preprocessing Image grey scaling, one-hot encoding, noise reduction filtering, and size reduction are all part of the pre-processing procedures that greatly improve the network's overall accuracy. In this work, the picture data is initially 400 × 100 pixels in size. According to our findings, the model's performance remains unchanged when the picture size is reduced to 200 × 50 pixels, yielding almost identical outcomes. Because it decreases the data without requiring any deduction in the data entropy, this size reduction may greatly assist the training process in becoming much quicker. Greyscale image processing was another preprocessing step. The pictures used for captchas were changed to grayscale. In doing so, we can keep the detection accuracy level while reducing the data size. Both the prediction and training processes are made easier, and the quantity of redundant data is decreased, thanks to this. Since color is not crucial for CAPTCHAs based on alphanumeric text, converting the source picture to greyscale has little effect on the outcomes. Normalization was the final preprocessing method we used. An picture is normalized when its pixel values are divided by the highest possible value for a pixel, which in this example was 255. By standardizing the data distribution of each input parameter, normalization speeds up the model. As a result, the model's convergence during training is accelerated. Subject: B. One-Hot Encoding The number of classes relies on the amount of alphabetical letters and numerical digits when dealing with CAPTCHAs, unlike other classification problems where numerous classes need to be predicted. The duration of the CAPTCHAs is another factor to consider. For this project, we have used a five-character CAPTCHA. Depending on the amount of classes that need to be discovered, this leads to exponential growth. So far, we have explored around 100,000 distinct permutations for a CAPTCHA issue involving five number digits. Consequently, we can only use one neural network, thus we have to encode the output data accordingly. Our data set includes one hot-encoding that was created using the image CAPTCHA module in Python. A machine learning technique is transforming category input into binary numbers in a single hot encoding. We chose 26 letters and 10 digits to make the predictions more accurate. One of the 36 possible combinations may be used for each CAPTCHA unit. The first letter is encoded with 1, while the remaining 35 combinations are encoded with 0, for instance, if the first letter is B. In the same way, all four CAPTCHA modules have been completed. The input data is categorical in nature, thus it's important to employ

International Journal of Engineering Research and Science & Technology

ISSN 2319-5991 www.ijerst.com

Vol. 21, Issue 2, 2025

one-hot encoding to ensure accurate prediction. Thus, our model is able to make more accurate predictions with the aid of encoding. Algorithm 1 provides the algorithm that is utilized for one-hot encoding.

C. ONE HOT ENCODING ALGORITHM: Algorithm 1

$targs \leftarrow np.zeros((5, num_symbols))$;
$pictarget \leftarrow 5$;
while $j, l \neq pictarget$ **do**
  $ind \leftarrow symbols.find(l)$;
  $targs[j, ind] \leftarrow 1$;
end
$X[i] \leftarrow img$;
$y[:, i] \leftarrow targs$;

D. Model Architecture Despite this, RNN is a viable solution for CAPTCHA prediction and cracking. Here, we present the results of using sequential CNN. model to bypass CAPTCHA, as they can do so more quickly and, with good design, provide better results than RNNs. Figure 5 shows the architecture of the network that we have suggested. Beginning with a convolutional layer containing 16 input neurons, the 3×3 Kernels, and the ReLU activation function (1), the network gets going. After this layer, there is a 2 × 2 Max-Pooling module. Two sets of convolutional, batch normalisation, and Max-Pooling layers with 32 neurons each and the ReLU activation function follow these layers (1). It should be mentioned that the "same" padding parameter is used by all the convolutional layers. Following these layers, a flatten layer is used to merge all the 2-dimensional arrays that are collected from the pooled feature maps into one long continuous linear vector. This flatten layer now splits into five separate branches, one for each of the five alphanumeric characters used in the CAPTCHA. A dense-dropout-dense layer is present in every branch. A total of 64 neurons are activated. There are two dense layers: one with an activation function called ReLU and another with an activation function called

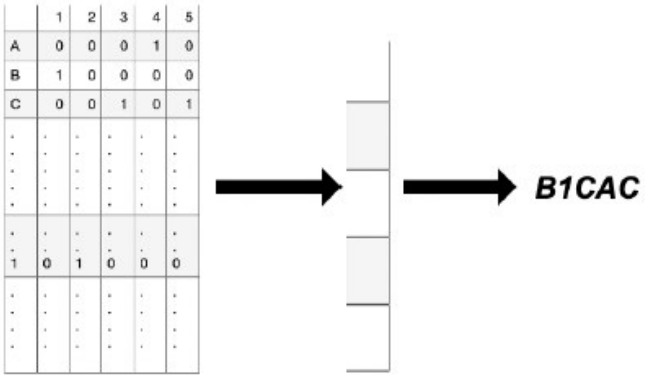


Fig. 4: One-hot encoding

International Journal of Engineering Research and Science & Technology

ISSN 2319-5991 www.ijerst.com

Vol. 21, Issue 2, 2025

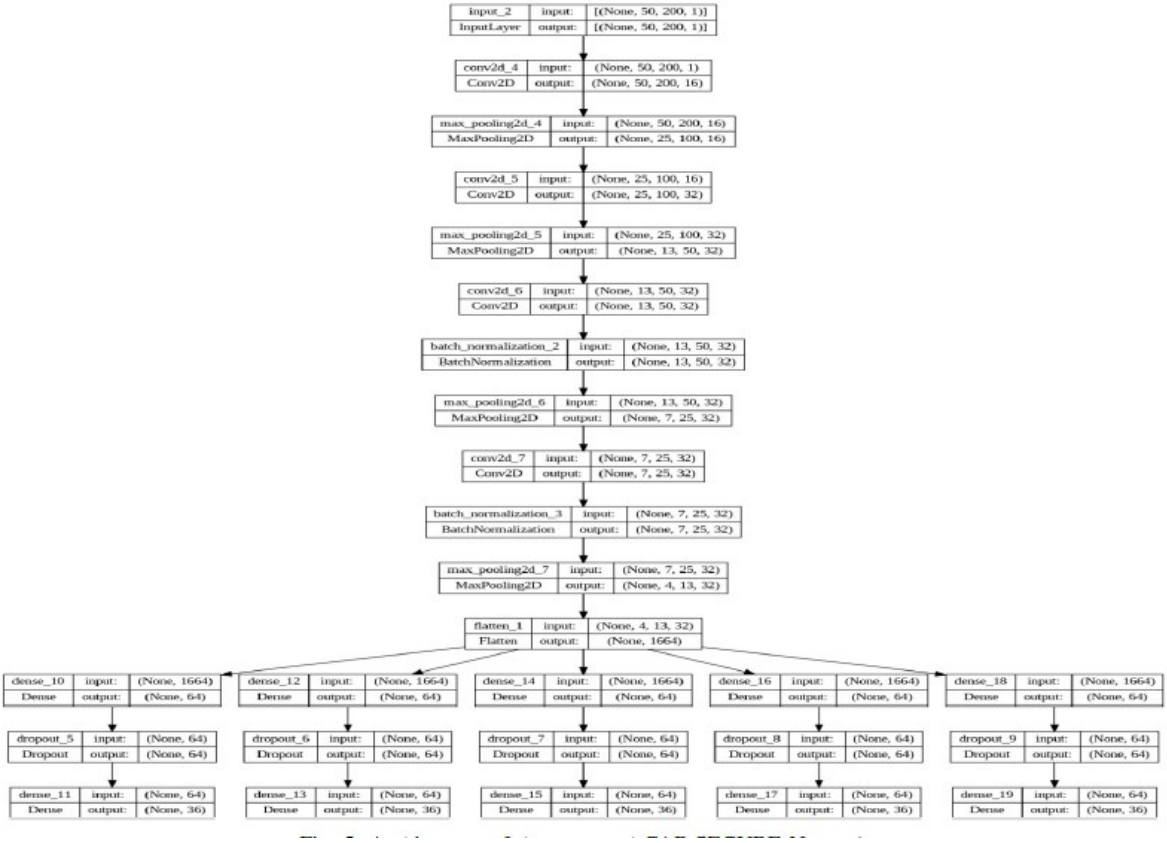


Fig. 5: Architecture of the proposed CAP-SECURE Network

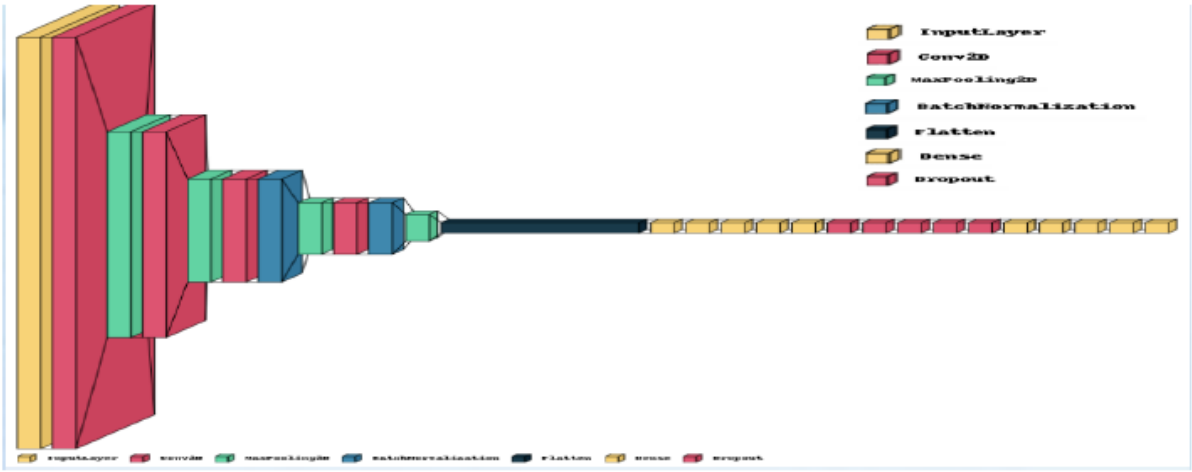


Fig. 6: 3D view of the proposed CAP-SECURE Network

International Journal of Engineering Research and Science & Technology

ISSN 2319-5991 www.ijerst.com

Vol. 21, Issue 2, 2025

set to 0.5 which is 50%. The ReLU (f(x)) and the Sigmoid activation (σ(x)) functions are mathematically expressed as

f(x) = max(0, x), (1)

and

σ(x) =

1

1 + e−x (2)

respectively. The loss function (3) of the proposed network is the Binary cross entropy as we need to compare these binary matrices together.

$$\sum_{i=1}^{N} -(y_i \log(p(x_i)) + (1 - y_i) \log(1 - p(x_i)),$$

where xi and yi are the input CAPTCHA for the ith sample, and N is the number of samples. Due to the fact that label could be either 0 or 1, the binary cross entropy is used. The equation would only be active in part. Equations (4) through (8) outline our model, which makes use of the ADAM optimizer. Here, m(t) and v(t) stand for the sum of squared previous gradients and the aggregate of gradients at time T, respectively.

$$m_t = \beta_1 \cdot m_{t-1} + (1 - \beta_1)[\delta(L)/\delta(w_t)],$$

and

$$v_t = \beta_2 \cdot v_{t-1} + (1 - \beta_2)[\delta(L)/\delta(w_t)]^2$$

in that order. In this context, β1 and β2 are adjustable constants that are also called moving average parameters. The optimizing function's gradient is denoted as δ(L)/δ(wt), and the learning iteration is denoted by t. Here are the momentary values of m and v as mentioned in Equations (6) and (7):

$$\hat{m}_t = m_t/(1 - (\beta_1)^t).$$

$$\hat{v}_t = v_t/(1 - (\beta_2)^t).$$

The best value of the function is finally obtained by computing the value of θt using equation (8). Eqs. (6) and (7) are used to determine ˆmt and ˆvt. "η" as well as "known" step size since our method uses a learning rate of.001.

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\hat{v}_t} + \epsilon}\hat{m}_t.$$

International Journal of Engineering Research and Science & Technology

ISSN 2319-5991 www.ijerst.com

Vol. 21, Issue 2, 2025

Using ADAM optimizer allows for the network to be trained in an acceptable amount of time. Its rapid convergence yields better results than SGD. The In Figure 7, the outcomes are contrasted. Our model was trained for 200 epochs with 32-batch sizes after many tests. The network continues to show signs of satisfactory convergence even after 100 epochs, as seen in Figure 7. The network seems to be able to maintain a steady performance after 200 epochs. We used the Python Image Captcha Library to train our model on 200,000 randomly generated CAPTCHAs after we developed the model mentioned above [31]. Some of the numerical CAPTCHAs that are created at random and have a set length of five digits are shown in Figure 2.

IV. EXPERIMENTAL RESULTS

Section A. Evaluation of Results Our model is compared to several well-known CNN models and previous research in this field in this section. When comparing The findings are shown in Table I, and subsequent sections address the methods' specifications. Our method relies on CNN, as mentioned before, which has

TABLE I: Accuracy metric of the dataset for each digit and the complete CAPTCHA as a set of 5 integrated digits.

	Train	Test
Digit1	99.4%	99.4%
Digit2	98.9%	94.3%
Digit3	98.7%	94.3%
Digit4	97.03%	94.3%
Digit5	98.19%	97.9%
CAPTCHA	96.45%	94.67%

TABLE II: Loss and Accuracy of the Model

	LOSS	ACCURACY
Train	.023	96%
Test	.095	94.67%

First, two layers of Convolutional Batch Normalization- MaxPool, and then two pairs of Convolutional-MaxPool layers .After going through, it splits into five branches. deflate the layer. The last step is to train the network using Adam optimiser. Batch normalization allowed us to rescale the data to a mean of 0 and standard deviation of 1 for our network, making it quicker and more reliable. This eliminates the negative impact of the internal covariance shift and helps us attain a fixed distribution of inputs. Figure 8 displays the results of our model's predictions for CAPTCHAs extracted from several websites. But as many other approaches are compatible with both numerical and alphanumerical CAPTCHAs, we created a separate network that can solve either kind. Over two hundred thousand alphabetical CAPTCHAs were used to train the network. Two well-known CNN architectures, VGG and Alex-net, were used to compare our findings. The accuracy rates for both training and testing were 98.4 and 87%, respectively, for VGG-16. Alex-Net achieved 95% accuracy during training and 85% accuracy during testing. Compared to Alex-Net and VGG-16, our model performs better. Just like TOD-CNN [10], which uses a CNN model trained on a 60,000-item dataset, has also employed segmentation techniques to detect the characters. For picture and character segmentation, the method employs a TensorFlow Object Detection (TOD) approach. It was 92.37 percent accurate. B. Analyzing Vulnerabilities We were able to glean a few insights that would aid in the development of more reliable CAPTCHAs after meticulously visualizing the misclassified ones. It is possible to make the CAPTCHA generator less susceptible by paying attention to a few key elements. The model's misclassified CAPTCHAs would be easy for a normal person to see. It is possible to make more resilient CAPTCHAs by keeping a few things in mind.

International Journal of Engineering Research and Science & Technology

ISSN 2319-5991 www.ijerst.com

Vol. 21, Issue 2, 2025

1) Compared to the average intensity, the grey scale intensity was low in 86% of the misclassified CAPTCHAs.

2) The numbers 8 and 3 were the cause of misclassification in 58% of the instances.

3) Rotation of 12 degrees or more was applied to the numbers or characters in 80% of the situations.

4) The relationship between the numbers 1 and l, as well as the letters g and 8, was often muddled.

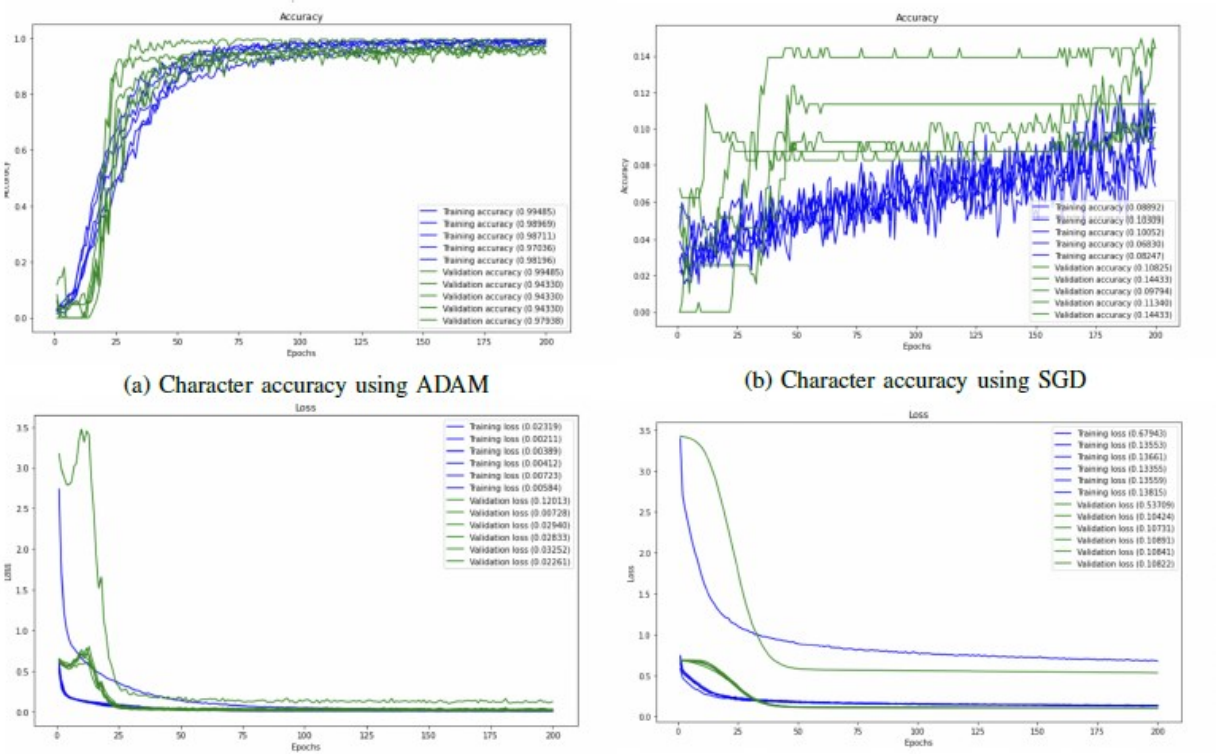


Fig. 7: Comparison of results between ADAM and SGD

To make CAPTCHAs more secure and resistant to machine and bot assaults, we recommend adding the numerals 8 and the characters g and l. Doing so will result in The CAPTCHAs are easy for humans to decipher, but machines have a hard time doing so.

V. CONCLUSION AND EXTENSION

In order to determine the susceptibility of CAPTCHAs used by popular websites, we developed a custom-based NN model capable of cracking alphanumeric CAPTCHAs. electricity producers. The model became more stable and quicker after using two batch normalization layers. It was useful in ensuring precision

International Journal of Engineering Research and Science & Technology

ISSN 2319-5991 www.ijerst.com

Vol. 21, Issue 2, 2025

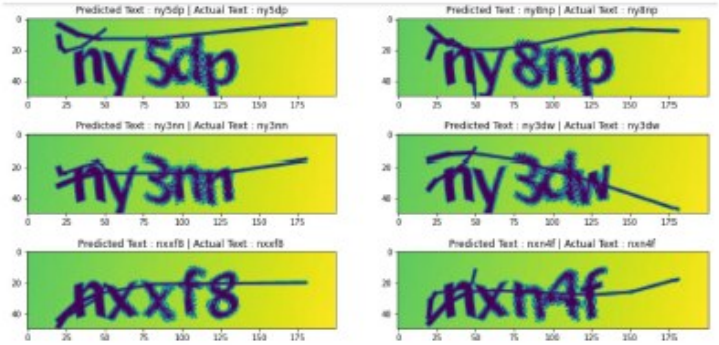


Fig. 8: CAP-SECURE model predicting CAPTCHAs taken

from random websites .surpassing state-of-the-art CNN models such as ALEXNET and VGG by a significant margin 94.67 percent). While our model performed well with random CAPTCHAs, it is worth noting that there are very few CAPTCHAs made CAP-SECURE network operations complex and hard. The purpose of studying such misclassified CAPTCHAs was to develop more secure CAPTCHAs that bots would have a hard time cracking. Fixing CAPTCHAs of varying lengths might be a future expansion route. The functionality may be expanded to accommodate CAPTCHAs of other languages. Additionally, future research may potentially investigate the use of image-based CAPTCHAs.

REFERENCES

[1] M. A. Kouritzin, F. Newton, and B. Wu, “On random field completely automated public turing test to tell computers and humans apart generation,” IEEE Transactions on Image Processing, vol. 22, no. 4, pp. 1656– 1666, 2013.

[2] B. B. Zhu, J. Yan, G. Guanbo Bao, M. Maowei Yang, and N. NingXu, “CAPTCHA as graphical passwords-a new security primitive based on hard AI problems,” IEEE Transactions on Information Forensics and Security, vol. 9, no. 6, pp. 891 – 904, 2014.

[3] Bostik, Ondrej, and Jan Klecka. “Recognition of CAPTCHA characters by supervised machine learning algorithms.” IFAC-PapersOnLine 51, no.6 (2018), pp. 208 – 213.

[4] Yousef, Mohamed, Khaled F. Hussain, and Usama S. Mohammed. “Accurate, data-efficient, unconstrained text recognition with convolutional neural networks.” arXiv preprint arXiv:1812.11894 (2018).

[5] Sivakorn, Suphannee, Jason Polakis, and Angelos D. Keromytis. “I’m robot: deep learning to break semantic image captchas.” In 2016 IEEE European Symposium on Security and Privacy (Euro S&P), 2016, pp.388 – 403.

[6] Von Ahn, Luis, Benjamin Maurer, Colin McMillen, David Abraham, andManuel Blum. “recaptcha: Human-based character recognition via web security measures.” Science 321, no. 5895, 2008, pp. 1465 – 1468.

[7] M. Belk, C. Fidas, P. Germanakos, and G. Samaras, “Do human cognitive differences in information processing affect preference and performance of CAPTCHA?” International Journal of Human-Computer Studies, vol.84, 2015, pp. 1 – 18.

[8] I. J. Goodfellow, Y. Bulatov, J. Ibarz, S. Arnoud and V. Shet, “Multi-digit Number Recognition from Street View Imagery using Deep Convolutional Neural Networks,” Computer Science, 2013.

International Journal of Engineering Research and Science & Technology

ISSN 2319-5991 www.ijerst.com

Vol. 21, Issue 2, 2025

[9] M. Jaderberg, A. Vedaldi, and A. Zisserman, "Deep features for text spotting," European conference on computer vision, Springer, Cham, 2014, pp. 512 – 528.

[10] K. Chellapilla and P. Y. Simard, "Using machine learning to break visual human interaction proofs (HIPs)," Advances in Neural Information Processing Systems, 2004, pp. 265 – 272.

[11] J. Yan and A. S. El Ahmad, "Breaking visual CAPTCHAs with naive pattern recognition algorithms," in Proceedings of the 23rd Annual Computer Security Applications Conference, Miami Beach, FL, USA,December 2007, pp. 279 – 291.

[12] H. Gao, J. Yan, F. Cao et al., "A simple generic attack on text captchas,"in Proceedings of the Network & Distributed System Security Symposium, San Diego, CA, USA, February 2016, pp. 220 – 232.

[13] G. Mori and J. Malik, "Recognizing objects in adversarial clutter: Breaking a visual CAPTCHA," Computer Vision and Pattern Recognition, Proc. IEEE Computer Society Conference, Vol. 1, IEEE, 2003, pp.I-I.

[14] K. Qing and R. Zhang, "A multi-label neural network approach to solving connected CAPTCHAs," in Proceedings of the 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR), IEEE Computer Society, Kyoto, Japan, November 2017, pp. 1313– 1317.

[15] B. Shi, X. Bai, and C. Yao, "An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition," IEEE Transactions on Pattern Analysis & Machine Intelligence, vol. 39, no. 11, pp. 2298 – 2304, 2016.