Computers, Materials & Continua
DOI: 10.32604/cmc.2022.024221
Article

Tech Science Press

Your CAPTCHA Recognition Method Based on DEEP Learning Using MSER Descriptor

Deepak Kumar[1], Ramandeep Singh[2] and Sukhvinder Singh Bamber[3,*]

[1]Faculty of Science and Technology, Lovely Professional University, Phagwara, 144411, India
[2]Lovely Professional University, Phagwara, 144411, India
[3]Department of Computer Science and Engineering, Panjab University SSG Regional Centre, Hoshiarpur, 146021, India
*Corresponding Author: Sukhvinder Singh Bamber. Email: ss.bamber@gmail.com

Received: 09 October 2021; Accepted: 06 January 2022

Abstract: Individuals and PCs (personal computers) can be recognized using CAPTCHAs (Completely Automated Public Turing test to distinguish Computers and Humans) which are mechanized for distinguishing them. Further, CAPTCHAs are intended to be solved by the people, but are unsolvable by the machines. As a result, using Convolutional Neural Networks (CNNs) these tests can similarly be unraveled. Moreover, the CNNs quality depends majorly on: the size of preparation set and the information that the classifier is found out on. Next, it is almost unmanageable to handle issue with CNNs. A new method of detecting CAPTCHA has been proposed, which simultaneously solves the challenges like preprocessing of images, proper segmentation of CAPTCHA using strokes, and the data training. The hyper parameters such as: Recall, Precision, Accuracy, Execution time, F-Measure (H-mean) and Error Rate are used for computation and comparison. In preprocessing, image enhancement and binarization are performed based on the stroke region of the CAPTCHA. The key points of these areas are based on the SURF feature. The exploratory outcomes show that the model has a decent acknowledgment impact on CAPTCHA with foundation commotion and character grip bending.

Keywords: Captcha; MSER; ANN; SURF; SNN

1 Introduction

CAPTCHAs, commonly known as "Completely Automated Public Turing test to tell Computers and Human Apart" tests are used to distinguish computer programs and human beings. These CAPTCHAs also provide security against bots. Several CAPTCHAs such as: image based, text based, audio based, and video based have been introduced in the last few years, but these are still defenseless in fighting against security. However, text-based CAPTCHAs are quite popular owing to its easy implementation and robust feature [1]. Many researchers developed various methods to protect text-based CAPTCHAs which includes: random lines, background noise, text distortion, merging of characters and many more. Generally, CAPTCHA being a security mechanism has been designed to defend against attacks on single-layer text-based CAPTCHAs designed by yahoo, google and Amazon [2]. Further, Microsoft in 2015, installed a two-layer Captcha scheme and achieved a success rate of 44.6% on a standard desktop version of personal computer Now, scientists have begun examining mechanized techniques to settle CAPTCHAs. It is widely believed that Text-based CAPTCHAs are mostly used because of low-speed attacks, particularly, 2–5 s per image, thus not a serious attack. The authors in [3] introduced an attack on text CAPTCHAs that successfully challenged that theory. The authors further claimed that using deep learning methods, the attack proposed by them successfully breached the Roman-character-based text CAPTCHAs deployed by the top global websites along with the three Chinese CAPTCHAs containing larger text sets. Similarly, authors in [4] also introduced effective deep learning scheme to attack newly emerged 11 Chinese CAPTCHAs which through their proposed attack achieved 34.7% to 86.9% success rate at speed of 0.175 s per image. Thus, it can be said that, text-based CAPTCHAs having larger character sets are also insecure. [5] presented yet another text captcha solver which they claimed to have higher efficiency of 0.05 s per image using a personal computer that is better in comparison to other existing methods. Besides the above few methods of text-based attacks, image restoration issue is a major issue in processing an image. Nowadays, discriminative CNN-based mechanisms are getting considerable attention because of their higher efficiency. The authors in [6] presented modern character segmentation method to attack a number of text CAPTCHAs, which includes the techniques deployed and designed by various multinational companies. They estimated that the CAPTCHAs can be instantaneously broken by a malicious bot with a success percentage of greater than 60%. Further in [7], the authors developed effective techniques based on shape context matching which recognize the words in EZ-Gimpy image with 92% accuracy. Also, network design architecture remains a challenge the authors in [8] proposed a denoising-based image restoration algorithm which uses an iterative step to unfold it into a deep neural network (DNN), which is composed of multiple denoisers modules interleaved with back-projection (BP) modules that ensure the observation consistencies. Further, a multi-scale redundancy of natural images is proposed that exploits the powerful denoising ability of DNNs.

Further, the authors in [9], the division results were recognized by KNN classifier, back proliferation (BP) system and bolster vector machine (SVM) individually, and the acknowledgment rates were all above 95%. Next, the authors in [10] merged region proposal network and fast R-CNN into a single network in which the region proposal network module states the unified network where to look. For VGG-16 model, this proposed detection model sets a frame rate of 5fps on personal computer. Further, with an increasing training data and machine performance improvement, an object detection technique algorithm based on CNN has been used by many researchers, but owing to its complex background and low resolution, issues still persist in detecting smaller object. Thus, in order to overcome such issues, the authors in [11] proposed faster region-based CNN for detecting smaller objects. [12] proposed segmenting Chinese license plates (LPs) in which instead of using old digital image processing methods, an innovative license plate segmentation is executed via powerful deep learning methods. In the first step, preprocessing of input license plate image is performed followed by segmentation of license plate and separation of characters using CNN such that every character becomes an image. It is less complex and a good choice for some old digital image processing license plate segmentation techniques. Moving ahead, the authors in [13] proposed an effective CNN architecture based on Darknet for performing character recognition. In this work, features of input character images are firstly extracted with the help of max pooling and convolutional layers and are then forwarded for classification to softmax layer. Dropout regularization method is also opted for avoiding issue of overfitting and dataset of 84,000-character images is taken for training and testing the model. In [14], the scientists worked on for training a large and deep CNN to categorize

1.2 million high-resolution images into 1000 distinct classes. They further said that their approach is highly efficient in comparison to previous state-of-the-art. The authors also claimed that the proposed mechanism used dropout method to overcome problem of overfitting in much more efficient way. In the next research work, [15] described a learning method based on training CNN for classifying traffic sign system. [16] also proposed a segmentation technique of captcha based on improved vertical projection. The segmentation technique can effectively overcome the segmentation issue of distinct kinds of conglutination characters in CAPTCHAs. It combines the letters and numbers in order to improve the captcha's accuracy. Further, the probability pattern framework used in [17] was able to identified the target numbers in captcha images and their quantitative evaluation showed an average of 81.05%. [18–20] used LPs recognition systems for detecting a license plate of vehicles.

In [21], the authors tried to find the vulnerabilities in the CAPTCHAs and then recognition-based segmentation is implemented to ignore linked characters. The researchers in [22] primarily focused on evaluation of text-based Captcha images. They evaluated such CAPTCHAs with multiple noise degrees, size and skew along with the font type. In [23], the authors proposed a simple algorithm for LP reader for metropolitan cities in Bangladesh. They used "Mantra" as a feature of Bangla script that amalgamates the characters of words. [24] proposed an active deep learning method that makes helps to get new training data without any human involvement. [25] proposed an innovative method to solve CAPTCHAs without segmentation by means of multi-label CNN. [26] introduced an end-to-end attack on text-based CAPTCHAs via CNN and an attention-based recurrent neural network. [27] proposed a transfer learning-based technique which significantly minimized the complexity of an attack and its associated cost of labeling samples. [28] investigated few contemporary networks that presented an effect of the convolutional kernel depth, size, and width of CNN. [29] focused on recognition/detection of Chinese car LP in complex background. They further proposed the model to apply the YOLO detector for LP detection. The authors in [30] proposed a new neural network architecture that joins sequencing model, feature extraction, and transcription into a unified architecture. In [31], the authors have discussed the survey of methods for detecting corona virus and also the segmentation of lungs in the CT-scan and X-rays. In [32,33], hybrid model is deployed to improve the epileptic seizure detection of EEG signals. In [34], the researchers with the help of deep learning differentiated autism spectrum disorder. In [35], deep learning applications and methods for automated multiple sclerosis detection are studied and discussed. In [36], CNN auto encoder is proposed for predicting the survival of COVID-19 patients using a CNN training.

After going through the literature survey in detail, following techniques were chosen on the basis of robust and contemporary techniques which provided the maximum accuracy or success rates. The techniques are mentioned in Tab. 1 with its merits and demerits along with its success rate in percentage.

Table 1: Comparative analysis of techniques for breaking CAPTCHAs

Technique name	Reference	Advantages	Disadvantages	Success rate (%age)
Style area captcha	[3]	Improved security	Security is not ideal	10.1–90%
Style area captcha	[3]	Improved security	Security is not ideal	10.1–90%

(Continued)

Table 1: Continued

Technique name	Reference	Advantages	Disadvantages	Success rate (%age)
Character recognition convolutional neural network	[13]	satisfying classification ability and high recognition accuracy	Test accuracy not improved	99.69%
Automatic license plate recognition	[19]	Works best in worst case scenarios	Not suitable for mobile licence plate	84.8%
Recognition based segmentation	[21]	Results improved on making the dataset larger	Few highly overlapped characters were not recognized	95.5%
An end-to-end attack	[26]	Effectively broken hardest type of Google	Less user friendly	98.3%
Convolutional recurrent neural network	[30]	Better performance	Slow	95.9%

Although the above-mentioned techniques showed better performance and outperforms the existing techniques, but still there are many issues that needs to be resolved for higher accuracy or success rate goal. The success rate is not only the sole improvement criterion, other technical factors must also be taken care of while working with CAPTCHAs security.

1.1 Limitation of Study

Several limitations or disadvantages have been studied after going through the detailed analysis in the research area of CAPTCHA recognition as discussed below:

1. Security is not ideal
2. Slow
3. Less user friendly
4. Few highly overlapped characters were not recognized
5. Not suitable for mobile licence plate
6. Test accuracy not improved

These limitations can be improved if deep learning-based captcha recognition using MSER descriptor approach is followed (The proposed method).

In the initial hypothesis, all the depth study of the previous research work, it reveals that though a lot of research work has been done on text detection and recognition in natural scene images but most of the researchers have concluded their survey either on horizontal or near to the horizontal texts. Their survey somewhat speaks about multi orientation text detection but the curved text detection on natural images escaped their attention. It has necessitated exploring on the gigantic and vital aspect of text detection field where detailed study on horizontal, near to horizontal, multi orientation and curved text finds place in a single cover. To achieve this, we have designed a text detection system that increases the accuracy of horizontal, near horizontal and curved text than the previous research.

2 Datasets with its Features

A PyCaptcha has been used in this research paper which is a python package for CAPTCHA generation for making custom image dataset out of it. Several degrees of freedom are offered in the package including: font style, noise, and distortion. The main aim for exploiting the freedom is to enhance the diversity of data and its recognition. In the initial step, single-letter CAPTCHA images (40-by-60 pixels) have been created using PyCaptcha uppercase letters (A to Z) from a restricted font set and corresponding letters were used for labelling the resulting images. Therefore, in this way, a supervised classification problem with 26 classes has been formulated. Further, for actual CAPTCHA breaker, mapping of an image into a string of letters is done. And then a four-letter CAPTCHA image (160-by-60 pixels) dataset is also generated. However, due to limit on the number of classes, this research work has not considered each distinct four-letter string as a label. The storage of full four-letter string is performed, but we used 26 single-letter labels only (as discussed in a later section) for our multi-letter CAPTCHA algorithms.

A four-letter CAPTCHA image shown in Fig. 1 is generated along with a set of 1580 single-letter and four-letter CAPTCHAs that came from a different library of fonts. It computes how well the proposed algorithms based on a restricted set of training data have truly learned the fundamental archetype of each letter. This is known as "prediction" dataset.

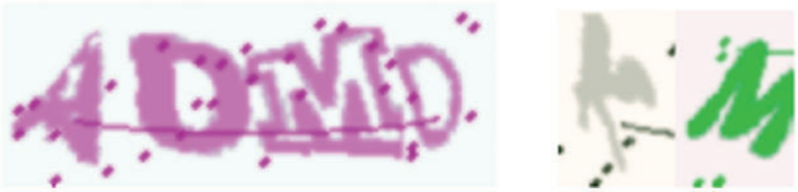

Figure 1: (Left) a typical CAPTCHA, which is an image distortion of the string "ADMD". (Right) a sample of two letters from the prediction dataset

3 Proposed Technique

Step 1: Training Samples are first input in an initial step.

Step 2: Then, in the second step, pre-processing is applied on the uploaded images.

3.1 Image Translation Based on Color

The image is changed to the gray image of single band after applying the color translation mechanism for uploading an image that assists in the text area segmentation with the use of MSER feature extraction (see Fig. 2).

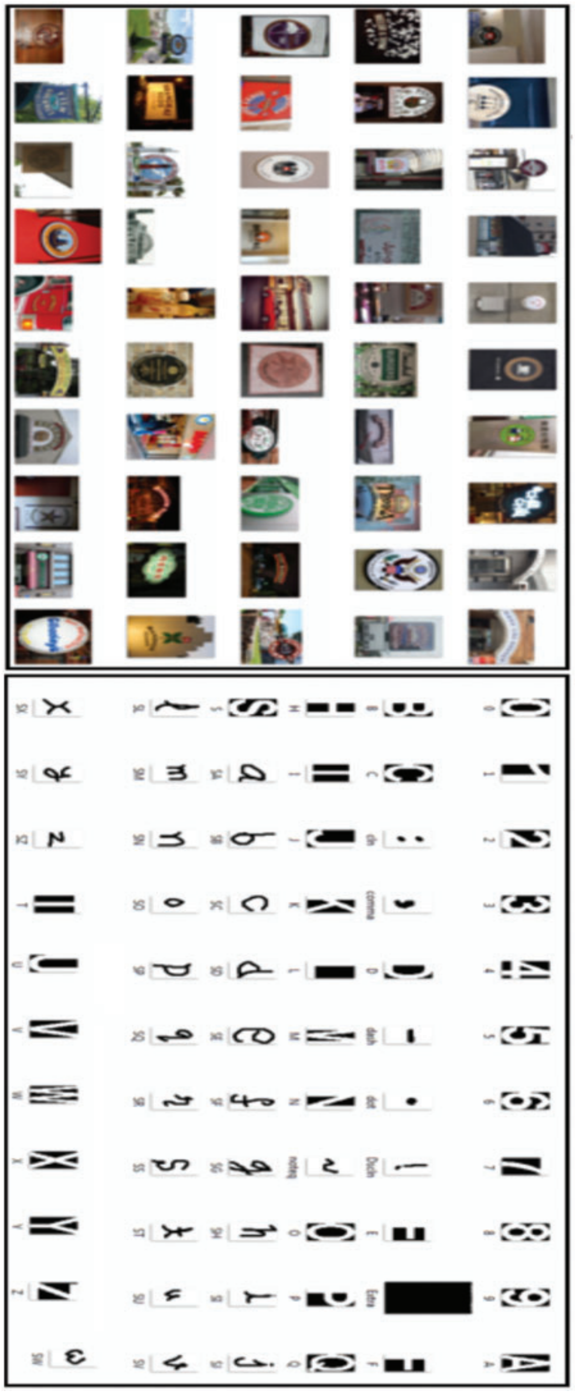

Figure 2: Sample containing 1500 dataset (CTW)

$$\text{GrayImage (GI)} = 0.299 \times R_c + 0.587 \times G_c + 0.114 \times B_c \tag{1}$$

Eq. (1) above has ‘R_c’ and ‘G_c’ as red and green component of image respectively and ‘B_c’ as image’s blue component.

Algorithm 1:

```
//Binarization
Input->I, Output->O and Average->AVG
I: (Gimg) is Gray Image and O: (Bimg is a Binary Image)
In Gimg, state Columns (CLM) and Row (RW)
{
for ($k = 1 \rightarrow RW$)
  {
    for ($m = 1 \rightarrow CLM$)
    If (Gimg (k, m) > AVG (Gimg))
{
Bimg (k, m) = 1
}
Elseif (Gimg (k, m) < AVG (Gimg))
{
Bimg (k, m) = 0
}
}
}
Return: Bimg as a BI
End
```

3.2 *Binarization*

A binarization method translates (GI) to (0, 1) form that is in binary form. It then helps in detecting the text region with ease.

Step 3: The non-linear collections of operation linked with the shape or morphology of image features, also known as morphological operation (see Fig. 3). Such operations are implemented by making use of few basic operations to binary images to locate an exact text region in image. Myriad morphological operations exist, but the morphological operations used in this research work are shown in Fig. 6. The figure illustrates the morphological operation practiced in the curved text detection model which is the proposed model. It helps in finding out deleted text regions and separate the regions with less text. Such morphological equations used are as follows:

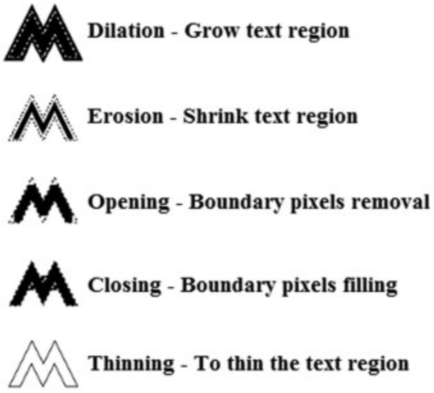
Dilation - Grow text region
Erosion - Shrink text region
Opening - Boundary pixels removal
Closing - Boundary pixels filling
Thinning - To thin the text region

Figure 3: Morphological operations

The area opening is performed in an image for pixel removing with an objective of discovering the good stroke section. The equation used for performing the opening is as follows:

$$\text{OpenImage} = \begin{cases} 0, & \textit{Pixels} < \text{Threshold (Acc to Otsu Algorithm)} \\ 1, & \textit{Pixels} \geq \text{Threshold} \end{cases} \tag{2}$$

3.3 Threshold

The boundary of the pixels in an image can be removed using the value of Pixels. An operation of morphological thinning is connected with a transformation of hit-and-miss. Otsu algorithm [37] is implemented in order to get the optimal value of threshold for every image. Additionally, BI (Bimg) thinning operation through 'X' element involves an outline as given:

$$\text{ThinImage } = \text{Bimg} - \text{Hit\&}\textit{Miss}\ (\textit{Bimg},\ X) \tag{3}$$

In (3), "Hit&Miss" is used for looking a linear pixel from BI, and 'X' being a structuring element of BI.

Step 4: In this step, a code developed for segmentation of text area is formulated by means of threshold-based segmentation, stroke detection, and MSER method from the pre-processed images.

Algorithm 2:

```
//MSER based Image Segmentation
Input-> I and Output-> O
I: Bimg stands for binary image and O: Simg is the segmented image
Outline Bimg (RW and CLM are the number of Rows and Columns)
{
For (k = 1→RW)
{
    For (m = 1→CLM)
{
Dimg (k, m) = DarkChannelExtraction (Bimg (k, m))
        Eimg (k, m) = ATMLightEnhancement (Bimg, Dimg (k, m))
        Timg (k, m) = TransmissionEstimate (Bimg (k, m), Eimg (k, m))
        Radiance (k, m) = GetRadiance (Eimg (k, m), Bimg (k, m), Timg (k, m))
        MSERReg (k, m) = MSERRegion (Bimg (k, m), RegionAreaRange (k, m))
}
}
Threshold 'Tr' is stated
For (k = 1→RW)
{
   For (m = 1→CLM)
{
    If (MSERReg (k, m)) lies under (Tr)
         {
Simg (k, m) = Considers as a Text Stroke
    }
```

(Continued)

Algorithm 2: Continued

```
            Simg (k, m) = Considers as Non-Text Stroke
    }
}
}
Return: According to text strokes, Simg as segmented image
}
Step 3: Extracting features
```

For extracting the features, SURF is used. An algorithmic illustration is performed using 3rd algorithm shown below:

Algorithm 3:

```
//SURF Feature Extraction
Input->I, Output->O and Scale Image as SI
I: (MSERReg)→(segmented image's text area)
O: (Key point(s))→(Features of text region)
State Columns (CLM) as number of and RW as number of Rows of MSERReg
{
For (k = 1→RW)
{
For (m = 1→CLM)
  {
   SI = Scaling (foreground (k, m), Scale Size)
      (L_Keypoint) = (Localization (Scale image (k, m))
       (O_Keypoint) = (Orientation (L_ Keypoint (k, m), Angle)
(SURF_Keypoints) = (Filtering (O_Keypoint (k. m), Square Filter))
} }
Return: (According to text strokes, SURF_Keypoint as feature of text area) }
Step 4: Classification and training by means of deep learning (DL)
```

3.4 Block Diagram

The block diagram of proposed work is given below in Fig. 4:

While designing the final methodology, we studied the following steps:-

a) Pre-processing algorithms (morphological operators)
b) Segmentation algorithms (MSER, SURF)
c) Various deep learning approaches such as Artificial Neural Network (ANN) and SVM also.
d) Performance metrics (Precision, recall & f-measures)

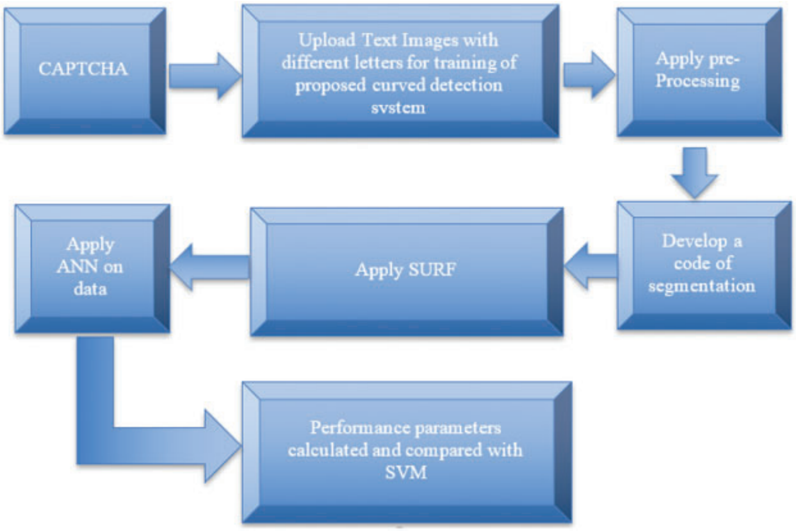
CAPTCHA
Upload Text Images with different letters for training of proposed curved detection svstem
Apply pre-Processing
Develop a code of segmentation
Apply SURF
Apply ANN on data
Performance parameters calculated and compared with SVM

Figure 4: Block diagram of the proposed work

4 Architecture of CNNs

One of the most imperative and significant in neural networks family is CNNs that are based on CPU for carrying out the image recognition task and text identification. The CNNs architecture is shown in Fig. 5 below. For CNN based curve text detection, an input natural image with text data inputs an image and processes it. Then, the classification is performed on it under certain categories (for example: A, B, C, D, E, F . . . . .Z). An input image is then considered as an array of pixel in CNN which is dependent on the resolution of image. Further, it considers H X W X D (H = Height, W = Width, D = Dimension) based on the resolution of input image it considers. Then, on the specific sections or areas of the image, a convolution is applied followed by pooling. Various layers of CNN architecture are shown below:

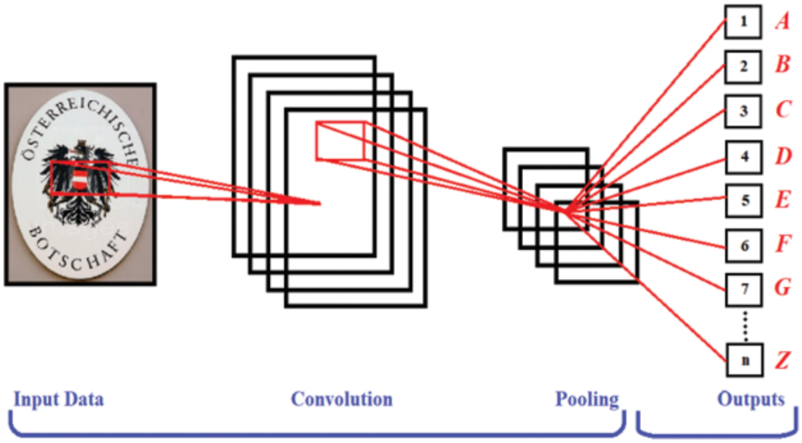

Figure 5: Architecture of CNN for curve text detection

Convolution Layer: It is a type of layer which extracts the features from natural input images in the CNN. It is also considered to be an initial layer for extracting characteristics from input natural images (NI). It is a layer that preserve the association/link between the pixels by learning the features of natural imaging by means of small squares (The red square box as shown in Fig. 6) of input data. A filter is used in a convolution layer for considering the pixels that are useful for training the system. Finally, all the residual pixels are considered as zero.

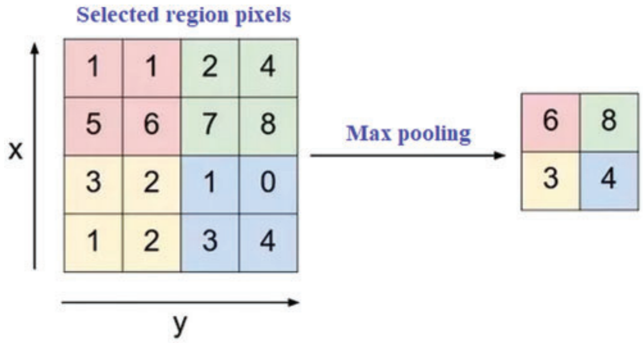

Figure 6: Method of max pooling method

Pooling Layer: The number of unwanted parameters is reduced using pooling layered in CNN. This happens when the size of the input NI is very huge and are not considered to be appropriate for target sets. There are different pooling methods such as: Sum Pooling, Max Pooling and Average Pooling. The proposed work consists of a max pooling method (as shown in Fig. 5) that is considered for training purpose. In max pooling technique, a largest set of pixels from the NI are corrected.

Output Layer (OL): For defining the target in accordance with the input data in CNN, OL is used. As depicted in figure where the architecture of CNN is shown, the outputs are considered as the text region (for illustration: A, B, C, D, E, F, G . . . .Z). An OL is responsible for generating the output of CNN which is then compared with an OL for producing the mean square error. Then, a performance function is expressed in a fully connected output layer to calculate the mean square error in accordance with the input data and a target. Further, for updating the filter (weights), a mean square error is back propagated and an addition of biased value is added for generating an output. Based on the training architecture or structure, now the proposed system can be tested after completing one training cycle of CNN. CNN further assists in classifying the text region from the NI (as shown in result section).

Algorithm 4:

```
//DL Algorithm
I->Input and O-> Output
I: (SURF_Keypoint)→(Feature of text area as a training data (TD)), (G)→(Target) and (NR)
→(Neurons); O: (CTRegion)→(Curve Text Region)
```

(Continued)

Algorithm 4: Continued

```
Initialize ANN with parameters: Epochs (E), Neurons (NR)
    • (Performance parameters): (MSE, Gradient, Mutation, and Validation Points)
    • (Training Techniques): (Levenberg-Marquardt (Trainlm)
    • (Data Division): (Random)

Outline RW and CLM (MSERReg)
{
for (j = 1)→ (all (TD))
     If TD lies in category 1 (Text Region)
{
Target (1, j) = 1
}
    Else
{        Target (2, j) = 1
}
}
}
Initialized the ANN using TD and Target
(Net) = (DL (TD, G, NR))
Set the training parameters according to the requirements and train the system
(Net) = (Train (Net, TD, G))
(CTRegion) = (simulate (Net, SURF_Keypoint))
Return: (CTRegion according to the curve text region)
}
```

The parameters used in the proposed work are: Precision, recall and F-Measure. In the precision, the number of positive class predictions is quantified which truly fit the positive class, while in recall the number of positive class predictions are performed out of all positive illustrations in dataset. Finally, F-Measure gives a single score which balances the concerns of recall and precision in a number. In proposed work, for producing output, the sigmoid function is used as activation functions. Nowadays, this function is widely being used in ANN for introducing non-linearity. Further, element of artificial neural network calculates input signal's linear combination and put on a function (sigmoid) to an output. Then a property is satisfied amid function derivative and itself, thus making it easy to calculate.

Step 4: The text region in the detection section is then formulated in accordance with the trained ANN architecture and mark it.

5 Experiments

The test equipment condition is Windows 10 64-piece, Inter(R) Pentium(R) CPU G3258, RAM 8 GB and the designs card NVIDIA GT980Ti. The product condition utilized in the trial is Tensor-Flow. During CAPTCHA, we expelled copy pictures to guarantee the dependability of the model. The size of pictures is 128∗48. The preparation set incorporate 5∗10^4 pictures, the approval set incorporate 2∗10^4 pictures and test set incorporate 1000 pictures. Fig. 7 gives a few instances of CAPTCHAs.

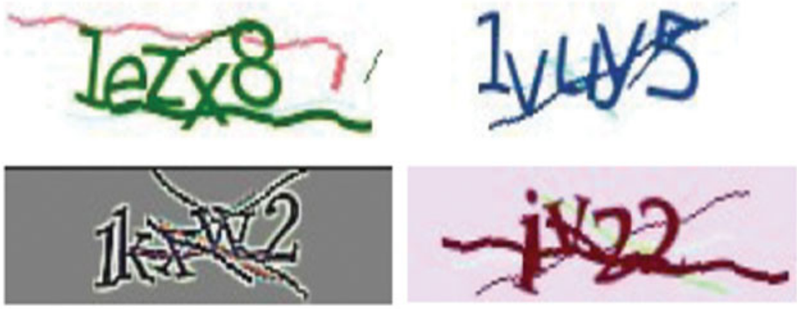

Figure 7: Example CAPTCHAs used in the experiments

The presentation assessment technique introduced by a strategy is exactness of an acknowledgment. In Tab. 2 below it is illustrated that the presentation of various techniques for CAPTCHA acknowledgment, together with SVM, BP neural system, and KNN calculations. In any case, each and every technique need to do preprocessing and fragmentation of pictures for accomplishing the reason for acknowledgment. The research likewise analyzes a presentation of an old-style convolution neural system LeNet to an equivalent dataset. Outcomes show that, contrasted and different strategies, the proposed technique has better acknowledgment execution.

Table 2: Performance assessment between method of CAPTCHA recognition

Technique/method	Accuracy of recognition
[6]	92%
[7]	60%
[9]	95%
LeNet	79.4%
Proposed technique	96.5%

6 Conclusion and Future Scope

The CAPTCHA is a testing technique used to recognize people and machines in organize condition. The investigations on CAPTCHA distinguishing proof can all the more likely identify vulnerabilities in the security of the CAPTCHA, in this manner forestalling some pernicious interruption in the system. The research performed in the paper proposed a CAPTCHAs recognition technology for finding out the exact regions in a text based on morphological operations. The exactness text region helps in the segmenting the regions of text and non-text regions. Further, using MSER, segmentation is performed for separating the text region with method of stroke width detection. Subsequently, in order to identify the key areas of segmentation, a SURF feature extraction is used. Exploratory outcomes show that the proposed strategy has a decent acknowledgment impact, and the acknowledgment precision ranges to 96.5%. Regardless of whether the plans of new CAPTCHAs can be secure are as yet open issues to be solved and are essential for future work.

Funding Statement: The authors received no specific funding for this study.

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

References

[1] R. Hussain, K. Kumar, H. Gao and I. Khan, “Recognition of merged characters in text based CAPTCHAs,” in *Proc. of the 2016 3rd Int. Conf. on Computing for Sustainable Global Development (INDIACom)*, 16–18 March, New Delhi, India, pp. 3917–3921, 2016.

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

[3] M. Tang, H. Gao, Y. Zhang, Y. Liu, P. Zhang *et al.,* “Research on deep learning techniques in breaking text-based CAPTCHAs and designing image-based captcha,” *IEEE Transactions on Information Forensics and Security*, vol. 13, pp. 2522–2537, 2018.

[4] P. Wang, H. Gao, Q. Rao, S. Luo, Z. Yuan *et al.,* “A security analysis of CAPTCHAs with large character sets,” *IEEE Transactions on Dependable and Secure Computing*, vol. 18, no. 6, pp. 2953–2968, 2021.

[5] G. Ye, Z. Tang, D. Fang, Z. Zhu, Y. Feng *et al.,* “Yet another text captcha solver: A generative adversarial network based approach,” in *Proc. of the 2018 ACM SIGSAC Conf. on Computer and Communications Security*, Toronto, ON, Canada, 15–19 October 2018, New York, NY, USA: Association for Computing Machinery, pp. 332–348, 2018.

[6] J. Yan and A. S. El Ahmad, “A Low-cost attack on a microsoft captcha,” in *Proc. of the 15th ACM Conf. on Computer and Communications Security (CCS ‘08)*, Association for Computing Machinery, New York, NY, USA, pp. 543–554, 2008.

[7] G. Mori and J. Malik, “Recognizing objects in adversarial clutter: Breaking a visual CAPTCHA,” in *2003 IEEE Computer Society Conf. on Computer Vision and Pattern Recognition, 2003. Proc.*, Madison, WI, USA, 2003, pp. 1, 2003.

[8] W. Dong, P. Wang, W. Yin, G. Shi, F. Wu *et al.,* “Denoising prior driven deep neural network for image restoration,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, pp. 2305–2318, 2019.

[9] J. Liang, “An approach to recognition of authentication code,” *Journal of Ningbo University*, n. pag., 2007. https://www.semanticscholar.org/paper/An-Approach-to-Recognition-of-Authentication-Code-Jun-li/d7650a2a3ce00ee8e8e83ba60754c3b4113831a1.

[10] S. Ren, K. He, R. Girshick and J. Sun, “Faster R-CNN: Towards real-time object detection with region proposal networks,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, pp. 1137–1149, 2017.

[11] Y. Liu, “An improved faster R-CNN for object detection,” in *Proc. of the 2018 11th Int. Symposium on Computational Intelligence and Design (ISCID)*, Hangzhou, China, 8–9 December, vol. 2, pp. 119–123, 2018.

[12] A. Abdussalam, S. Sun, M. Fu, H. Sun and I. Khan, “License plate segmentation method using deep learning techniques,” in *Proceedings of the Signal and Information Processing, Networking and Computers*, S. Sun, (Ed.), Singapore: Springer, pp. 58–65, 2019.

[13] A. Abdussalam, S. Sun, M. Fu, Y. Ullah and S. Ali, “Robust Model for Chinese License Plate Character Recognition Using Deep Learning Techniques,” In: Q. Liang, X. Liu, Z. Na, W. Wang, J. Mu, B. Zhang (eds.) *Communications, Signal Processing, and Systems. CSPS 2018. Lecture Notes in Electrical Engineering*, Singapore: Springer, vol. 517, pp. 121–127, 2020.

[14] A. Krizhevsky, I. Sutskever and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” *Communications of the ACM*, vol. 60, pp. 84–90, 2017.

[15] N. Jmour, S. Zayen and A. Abdelkrim, “Convolutional neural networks for image classification,” in *Proc. of the 2018 Int. Conf. on Advanced Systems and Electric Technologies (IC-ASET)*, Hammamet, Tunisia, 22–25 March, pp. 397–402. 2018.

[16] L. Zhang, Y. Xie, X. Luan and J. He, “Captcha automatic segmentation and recognition based on improved vertical projection,” in *Proc. of the 2017 IEEE 9th Int. Conf. on Communication Software and Networks (ICCSN)*, Guangzhou, China, 6–8 May, pp. 1167–1172, 2017.

[17] C. J. Chen, Y. W. Wang and W. P. Fang, “A study on captcha recognition,” in *Proc. of the 2014 Tenth Int. Conf. on Intelligent Information Hiding and Multimedia Signal Processing*, Kitakyushu, Japan, 27–29 August, pp. 395–398, 2014.

[18] Q. Wang, "License plate recognition via convolutional neural networks," in *Proc. of the 2017 8th IEEE Int. Conf. on Software Engineering and Service Science (ICSESS)*, Beijing, China, 24–26 November, pp. 926–929, 2017.

[19] N. Saleem, H. Muazzam, H. M. Tahir and U. Farooq, "Automatic license plate recognition using extracted features," in *Proc. of the 2016 4th Int. Symposium on Computational and Business Intelligence (ISCBI)*, Olten, Switzerland, 5–7 September, pp. 221–225, 2016.

[20] A. Sasi, S. Sharma and A. N. Cheeran, "Automatic car number plate recognition," in *Proceedings of the 2017 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS)*, Coimbatore, India, 17–18 March, pp. 1–6, 2017.

[21] R. Hussain, H. Gao, R. A. Shaikh and S. P. Soomro, "Recognition based segmentation of connected characters in text based CAPTCHAs," in *Proc. of the 2016 8th IEEE Int. Conf. on Communication Software and Networks (ICCSN)*, Beijing, China, 4–6 June, pp. 673–676, 2016.

[22] P. Sakkatos, W. Theerayut, V. Nuttapol and P. Surapong, "Analysis of text-based CAPTCHA images using template matching correlation technique," in *Proc. of the 4th Joint Int. Conf. on Information and Communication Technology, Electronic and Electrical Engineering (JICTEE)*, Chiang Rai, Thailand, 5–8 March, pp. 1–5, 2014.

[23] R. A. Baten, Z. Omair and U. Sikder, "Bangla license plate reader for metropolitan cities of Bangladesh using template matching," in *Proc. of the 8th Int. Conf. on Electrical and Computer Engineering*, Dhaka, Bangladesh, 20–22 December, pp. 776–779, 2014.

[24] F. Stark, C. Hazırba¸s, R. Triebel and D. Cremers, "CAPTCHA recognition with active deep learning," in *Proc. of the German Conf. on Pattern Recognition Workshop*, Aachen, Germany, 7–10 October, 2015.

[25] K. Qing and R. Zhang, "A Multi-label neural network approach to solving connected CAPTCHAs," in *Proc. of the 2017 14th IAPR Int. Conf. on Document Analysis and Recognition (ICDAR)*, Kyoto, Japan, 9–15 November, vol. 1, pp. 1313–1317, 2017.

[26] Y. Zi, H. Gao, Z. Cheng and Y. Liu, "An End-to-end attack on text CAPTCHAs," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 753–766, 2020.

[27] P. Wang, H. Gao, Z. Shi, Z. Yuan and J. Hu, "Simple and easy: Transfer learning-based attacks to text CAPTCHA," *IEEE Access*, vol. 8, pp. 59044–59058, 2020.

[28] M. Fu, N. Chen, X. Hou, H. Sun, A. Abdussalam *et al.,* "Real-Time Vehicle License Plate Recognition Using Deep Learning," in *ICSINC 2018: Signal and Information Processing, Networking and Computers*, S. Sun, (Eds.), Singapore: Springer, vol. 494, pp. 35–41, 2019.

[29] H. Sun, M. Fu, A. Abdussalam, Z. Huang, S. Sun *et al.,* "License Plate Detection and Recognition Based on the YOLO Detector and CRNN-12," in *Proceedings of the Signal and Information Processing, Networking and Computers*, S. Sun, (Ed.), Singapore: Springer, pp. 66–74, 2019.

[30] 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 and Machine Intelligence*, vol. 39, pp. 2298–2304, 2017.

[31] A. Shoeibi, M. Khodatars, R. Alizadehsani, N. Ghassemi, M. Jafari *et al.,* "Automated detection and forecasting of covid-19 using deep learning techniques: A review," arXiv preprint arXiv:2007.10785, 2020.

[32] A. Shoeibi, N. Ghassemi, R. Alizadehsani, M. Rouhani, H. Hosseini-Nejad *et al.,* "A comprehensive comparison of handcrafted features and convolutional autoencoders for epileptic seizures detection in EEG signals," *Expert Systems with Applications*, vol. 163, pp. 113788, 2021.

[33] A. Shoeibi, M. Khodatars, N. Ghassemi, M. Jafari, P. Moridian *et al.,* "Epileptic seizures detection using deep learning techniques: A review," *International Journal of Environmental Research and Public Health*, vol. 18, no. 11, pp. 5780, 2021.

[34] M. Khodatars, A. Shoeibi, N. Ghassemi, M. Jafari, A. Khadem *et al.,* "Deep learning for neuroimaging-based diagnosis and rehabilitation of autism spectrum disorder: A review," arXiv preprint arXiv:2007.01285, 2020.

[35] A. Shoeibi, M. Khodatars, M. Jafari, P. Moridian, M. Rezaei *et al.,* "Applications of deep learning techniques for automated multiple sclerosis detection using magnetic resonance imaging: A review,".arXiv preprint arXiv:2105.04881, 2021.

[36] F. Khozeimeh, D. Sharifrazi, N. H. Izadi, J. H. Joloudari, A. Shoeibi *et al.,* "Combining a convolutional neural network with autoencoders to predict the survival chance of COVID-19 patients," *Scientific Reports*, vol. 11, no. 1, pp. 1–18, 2021.

[37] Z. Wang and P. Shi, "CAPTCHA recognition method based on CNN with focal loss," *Complexity*, vol. 2021, pp. 1–10, 2021.