

ORIGINAL ARTICLE



Deciphering complex text-based CAPTCHAs with deep learning

Asadullah Kehar^{1*}, Rafaqat Hussain Arain¹, Riaz Ahmed Shaikh¹¹ Department of Computer Science, Shah Abdul Latif University, Khairpur, Pakistan OPEN ACCESS

Received: 03-04-2020

Accepted: 23-04-2020

Published: 16-05-2020

Editor: Dr. Natarajan Gajendran

Citation: Kehar A, Arain RH, Shaikh RA (2020) Deciphering complex text-based CAPTCHAs with deep learning. Indian Journal of Science and Technology 13(13): 1390-1400. <https://doi.org/10.17485/IJST/v13i13.126>

***Corresponding author.**

Asadullah Kehar

Department of Computer Science,
Shah Abdul Latif University,
Khairpur, Pakistan
asad.kehar@salu.edu.pk**Funding:** None**Competing Interests:** None

Copyright: © 2020 Kehar, Arain, Shaikh. This is an open access article distributed under the terms of the [Creative Commons Attribution License](https://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Published By Indian Society for
Education and Environment ([iSee](https://www.indjst.org/))

Abstract

Background: CAPTCHA is a mechanism to distinguish humans from bots. It has become standard means of protection from the misuse of resources on World Wide Web. Different types of CAPTCHAs are implemented but text-based schemes are the most widely used due to its easiness and robustness. A user is asked to type in the text from an image. The image is intentionally distorted to dodge the bots. Recognizing the text is easy for humans but very hard for computers. **Method/Findings:** In this work, a text-based CAPTCHA scheme with background clutter and partially connected characters is decoded. The main steps consist on preprocessing, segmentation and recognition. Several digital image processing techniques were applied during preprocessing, segmentation steps and convolutional neural network (CNN) was used for recognition process. Since massive data is required for CNN therefore data was generated synthetically. A complex text-based CAPTCHA scheme with varying number of letters: 3, 4 and 5 letters is decoded with the overall precision of 77.5%, 64.2% and 51.9% respectively.

Keywords: CAPTCHAs; HIPs; image processing; machine learning; CNN

1 Introduction

CAPTCHA (Completely Automated Public Turing test to tell Computers and Humans Apart) is a computer test-program meant to distinguish between a computer and human. CAPTCHAs are normally easily solvable by humans whereas current-state computers are incapable to decode them⁽¹⁾. Working on CAPTCHA dates back to 1997, when first CAPTCHA was invented⁽²⁾ at the DEC Systems Research Center for blocking malicious automatic submission of URLs to AltaVista website. CAPTCHA-solving has almost become standard to prevent abuse of online services by bots. The bots may send junk e-mails, post unauthorized advertisements and fill servers with heavy traffic. These misuses can decrease performance of internet servers. The text based CAPTCHAs are most popular because these are easy for most users and provide better security. In text-based CAPTCHA, a user is asked to type in noisy, distorted, string of random characters. Distortions are intentionally introduced in text-string to assure protection from bots. Different type of distortions: dots, lines, arcs, color characters, merged characters are introduced to clutter CAPTCHA.

Research has proved that computers are more efficient at recognizing single characters (regardless of transformation: scaling, rotation, translation)⁽³⁾. This implies that if the location of a particular character is known then recognizing the character for the computer is an easy task⁽⁴⁾. There is usability-difficulty tradeoff while designing the CAPTCHAs⁽⁵⁾. If the difficulty is increased then the string becomes difficult to read even for humans. Practically, balancing these opposite needs is difficult but required for good CAPTCHA. Decoding CAPTCHA involves two major tasks: Segmentation and Recognition. Segmentation is the process to locate and extract individual character, while Recognition refers to identifying segmented individual characters. Segmentation of non-connected characters is easier compared to connected characters. The applications of CAPTCHAs may include preventing spam e-mails, preventing spam comments in blogs, protecting web-site registration, protecting fake on-line polls to name a few. Human Interactive Proofs (HIPs) is yet another name for CAPTCHAs.

In⁽⁶⁾, Yan and Ahmed successfully decoded Microsoft's CAPTCHA. The method used to decode the CAPTCHA was pixel counts after performing some pre-processing. Chandavale and Sapkal⁽⁷⁾ has suggested a generic algorithm for measuring the strength of CAPTCHA by analyzing one of the parameters. The suggested algorithm can be used for segmenting the characters for various types of CAPTCHAs. The proposed algorithm is supposed to utilize the concepts of snake game, characters' projection value and patterns of touching the characters. Starostenko et al.⁽⁸⁾, presented a scheme to use three-color bar code along with heuristic recognition method. This method generated results with a segmentation accuracy of 56.55%, Recognition with a rate of 95% and overall success rate of 54.65 on reCAPTCHA, version 2011. Support-Vector Machine (SVM) classifier was used for recognition. Ahmed et al.⁽⁹⁾ proposed a technique to break Google's CAPTCHA version 2010. The method used to segment connected characters was shape patterns. The characters with dot shape, loop shape, s-shape and cross shape were subject of interest for the technique shape-patterns. The authors⁽¹⁰⁾ have conducted two user studies based on single character Recognition: The 1st user study investigates the precision of humans when different types of distortions (rotation, scaling, local and global warp) were applied. The 2nd user study investigates human precision on baseline by applying arc clutters of different densities.

In⁽¹¹⁾, Author designed a correlation algorithm. This algorithm could identify the word in EZ-Gimpy challenge image with a rate of 99% and the direct distortion estimation method can correctly decode the four letters in Gimpy-r challenge image with the rate of 78%. In⁽¹²⁾, the method used by Researchers' is based on two neural architectures along with two different feature extraction techniques: 1- Segmented Character Recognition and 2- Methodology used by researchers is basically to describe the neural-based character classifiers. Both Methods were investigated and experimentally used to acquire results.

R. Hussain et al.⁽¹³⁾, verified robustness of CAPTCHAs offered by local e-commerce sites of Pakistan. The CAPTCHAs offered by these sites were successfully decoded. The proposed algorithms achieved promising results on all attacked CAPTCHAs. Using thresholding methods, color filling segmentation, Recognition based segmentation methods and machine learning techniques; the said CAPTCHAs had successfully been decoded with an overall precision of up to 82.4%.

In⁽¹⁴⁾, the Author of the paper, introduced an algorithm that worked step by step to make the character understandable. Firstly, the noise was reduced and image was thinned through morphology method. Two parameters were determined from top and bottom as starting and ending or terminating point respectively. The largest gap from the top and bottom were considered as starting and ending point of the character. The free falling segmentation and tunnel segmentation was used for further clarification of the character.

In⁽¹⁵⁾, the methodology used by Researchers was based on two algorithms, 1- CHELLAPILLA'S Algorithm, included pre-processing image opening and labeling to defeat Yahoo's CAPTCHAs system, 2- PROPOSED SEGMENTATION Algorithm, intended to improve the success rate of segmentation. Both methods were investigated and experimentally used to acquire results. R. Hussain et al.⁽¹⁶⁾ presented a new segmentation and recognition method which practices basic image processing techniques comprising thresholding, thinning and pixel count techniques with an artificial neural network for text-based CAPTCHAs. The attacked scheme was CCT (Crowded Characters Together) based CAPTCHA. S. Karim et al.⁽¹⁷⁾ tested Shape detection algorithms to find accuracy in target detection and analyzed the processing time before implementing in such environment and the results showed optimal accuracy in matching weapons type with name and shape in predefined database. R. A. Shaikh⁽¹⁸⁾, suggested a new method to incorporate the spatial layout with primitive features for object recognition and explained that spatial relation or position of object helps to identify the target objects in the complex scene. In⁽¹⁹⁾, the region of interests extracted from more than thousand images through different angles to train machine for large-vehicle detection using RCNN, Fast-RCNN, Faster-RCNN and Cascade methods. A. A. Laghari et al.⁽²⁰⁾ established that reduction of luminance and chrominance has less impact on image quality as compared to resolution scaling which has greater impact on quality. In⁽²¹⁾, a novel training methodology which is based on compressed and down-scaled images is applied to reduce the training time. The training images are compressed at Quality Factor (QF) of 50 and down-scaled by scale factor of 0.5 to assess the performance for vehicle detection.

2 Data Generation

The new machine learning techniques have provided much better results than the traditional classification methods. Machine learning methods require huge data for training networks. Unfortunately no public source is available for such type of CAPTCHA related data acquisition, therefore data was generated synthetically. Complex type text-based CAPTCHAs (i.e. a text based string containing noise as well as partially connected characters) were created with their solutions as CAPTCHA labels. Three different types of text based CAPTCHAs were generated with varying number of characters: 3, 4 and 5 characters (as shown in Figure 1). As per number of characters, the length of 2D image is set as 180 pixels, 190 pixels and 200 pixels respectively while the width of all the three types of images was fixed at 80 pixels. The CAPTCHA letters were randomly selected of alpha numeric type: upper case letters (A-Z), lower case letters (a-z) and digits (0-9). Different types of noise: lines, crosses and other stuff were randomly introduced in CAPTCHA image to make it segmentation resistant.

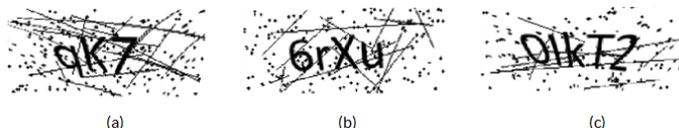


Fig 1. a) 3-Letter b) 4-Letter and c) 5-Letter complex text-based CAPTCHAs

3 Research Methodology

A novel framework is proposed (as shown in Figure 2) that follows the traditional approach to decipher complex text based CAPTCHAs involving preprocessing, segmentation and lastly the recognition. In preprocessing, due to different types of noise different methods are used: applying averaging filter, morphological operations and thresholding techniques. During segmentation step, dual segmentation is applied: compulsory segmentation to extract separable single characters and connected components/character. Optional segmentation is applied only for connected components to split it into its constituent single characters. For recognizing single characters, convolution neural network is used. The three steps are further discussed in the following section.

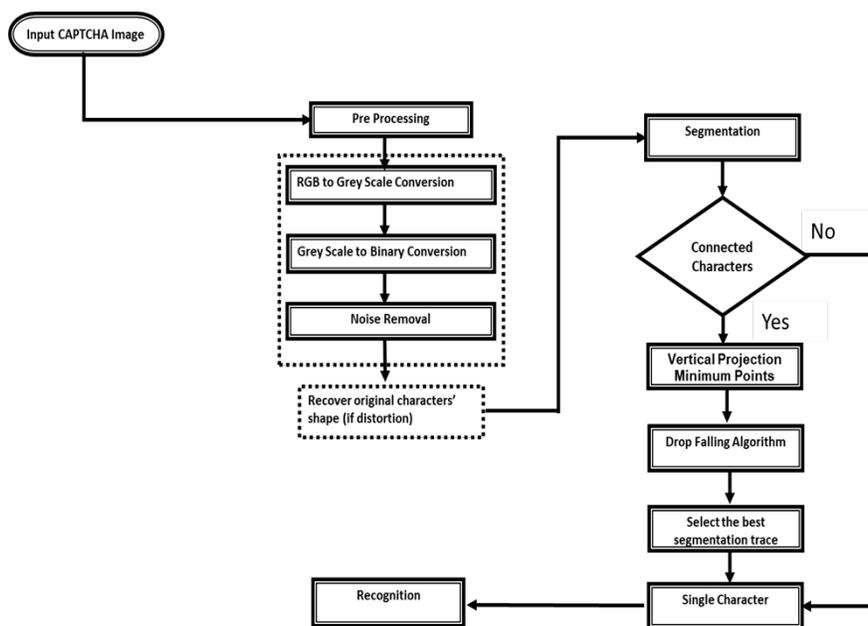


Fig 2. A novel framework for deciphering complex text-based CAPTCHAs

3.1 Preprocessing

All three versions of the generated CAPTCHA images contain digits: 0-9, lower case alphabets: a-z and upper case alphabets: A-Z. The main objective of preprocessing is to highlight the letters in the image and eliminate the interfering unnecessary details. Initially the image is converted in gray scale image (as shown in Figure 3a). To reduce the noise an averaging filter of kernel size 3X3 is applied on gray scale image. While applying averaging filter, the pixel intensity value is replaced by the mean value of its neighboring pixel intensities. After applying averaging filter, although irrelevant information is removed but images becomes blurred as well. The blurred image is shown in (Figure 3b). Image binarization is used to highlight the contour of text and remove the background noise. Selecting the appropriate threshold is the key to binarization and it is achieved by hit and trial method. Threshold range is set between minimum 110 and maximum 255. The binarized image is shown in (Figure 3c). After applying binarization technique further denoising is required. This time erosion technique is applied to eliminate noise of certain size and type with the help of structuring element. The Eroded image is shown in (Figure 3d).



Fig 3. a) gray scale image b) blurred image c) thresholded image and d) eroded image

Furthermore, after preprocessing steps, to recover original shape some repairing of steps are required often. While removing the arcs, lines and other types of noise, the original characters were also affected (i.e. cracks were created). Therefore, some mechanism was needed to restore the shape of characters. (Figure 4 a) shows the affected character and (Figure 4b) shows the recovered character. The fixation of broken characters can fill all one pixel wide gaps⁽²²⁾ and it works as follows:

1. Locate the pixels similar to background color and have left and right neighboring pixels similar to foreground color.
2. Locate the pixels similar to background color and have top and bottom neighboring pixels similar to foreground color.
3. Convert the color of above located pixels to the foreground color.

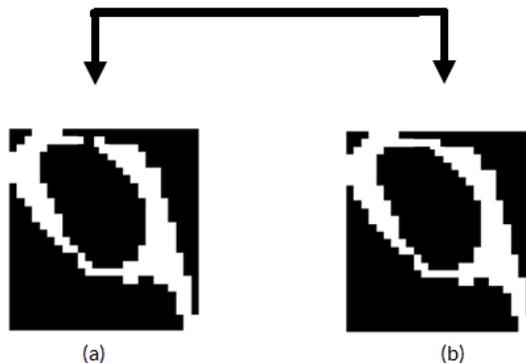


Fig 4. a) Affected character and b) Recovered character

3.2 Segmentation

The main objective of segmentation is to extract individual characters or connected components. The characters and connected components were extracted by considering connectivity. Connectivity refers to relationship between neighboring pixels. If the next left, right, up and down pixels of the current pixel are considered, it is referred as 4-connectivity and if the 4 pixels on the diagonal of current pixel are also considered then it is referred as 8-connectivity, as shown in (Figure 5). In this work individual characters and connected components were extracted by considering 4-connectivity. 80-85% individual characters and 15-20%

connected components were extracted during this phase (i.e. compulsory segmentation).



Fig 5. a) 4-connectivity b) 8-connectivity

The segmentation process started by initially omitting all the objects/characters which were less than 50 pixel-count. Only those objects were considered which fulfill the 4-connectivity criteria and those objects were also removed which did not meet the criteria. Moreover, the under consideration objects/characters were highlighted with the help of rectangle, as shown (Figure 6) and the highlighted characters were extracted and stored in folder named similar to the extracted character. Each folder contained multiple samples for a particular character as shown in (Figure 7)



Fig 6. CAPTCHA letters located and high lighted

These multiple samples of the individual characters are to be fed in CNN for training. The segmented objects contained 15% to 20% connected characters as shown (Figure 8). The decision whether the extracted object was single character or chunk of connected characters was based on the morphological analysis^(23–25).

The foreground pixel count and the width or number of columns occupied by each connected component were calculated to decide the number of characters contained by each connected chunk. Table 1 shows the morphological analysis done for deciding number of characters in connected components.

Table 1. Table 1: Morphological Analysis of CAPTCHA characters

Number of characters	1	2	3	4
Foreground pixel count	128-285	258-410	431-615	615-915
Columns occupied	15-25	30-38	45-59	62-93

The extracted single characters were already stored in relative folder to be fed for training and recognition but the connected components needed to be further segmented by drop-fall algorithm. The drop-fall algorithm was run after getting starting point by considering vertical projection minimum point^(26–28).

3.2.1 About drop-fall algorithm

A drop-fall algorithm was used to create an optimal segmentation path between two connected characters by mimicking a drop of water falling from top upon the contour of first connecting characters, as shown (Figure 9). In this algorithm, the hypothetical drop fell downward on the contour of a character until it blocked into a valley and then it would continue moving down, splitting through supposedly connected component.

The algorithm produced promising results even though it was very simple. The main challenge was the position where the drop would start falling from: if the drop starts falling from wrong position, it can be ineffective by either falling on left side

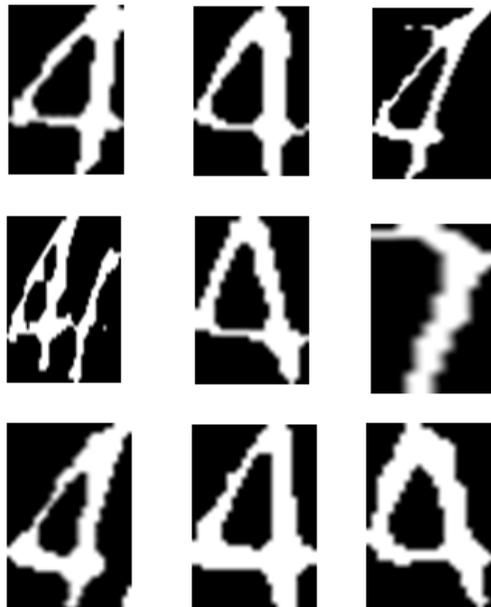


Fig 7. Multiple samples of characters



Fig 8. Some connected characters

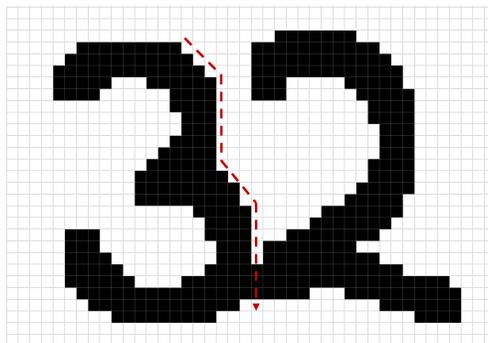


Fig 9. Drop-fall segmentation path

of the first character or right side of the second character. The best position to start algorithm is the closest point where two characters are joined together.

In drop-fall algorithm, pixels are traversed row by row (from left to right) until a black border pixel is encountered with another black border pixel on the right side. These black pixels are separated by white pixels. The location next to this encountered pixel is used as the starting point for drop fall segmentation path, as shown in (Figure 10).

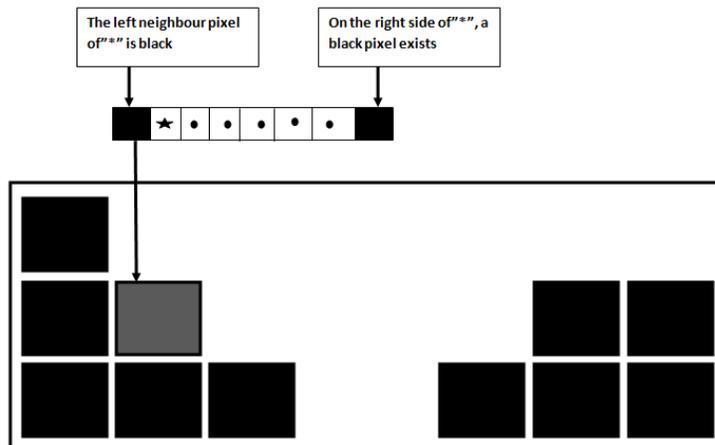


Fig 10. Seepage starting point of drop-fall algorithm

Since the drop fall algorithm mimics the falling of drop, hence it will always move down ward diagonally: either to left of character or to the right of character. The direction of moving depends on the current position of pixel and surrounding pixels. Drop-fall moving rules are given in Figure 11.

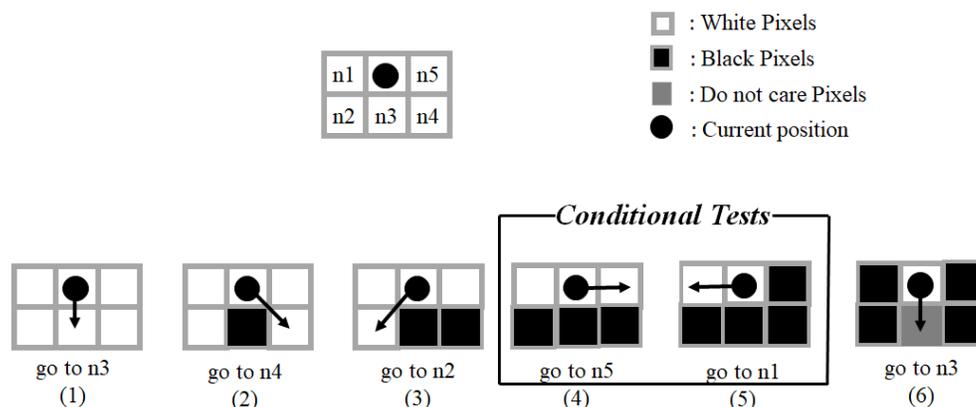


Fig 11. Drop-fall moving rules

- Rule #1: if n3=0, then nextPixel = n3
- Rule #2: if n4=0, then nextPixel = n4
- Rule #3: if n2=0, then nextPixel = n2
- Rule #4: if n5=0, then nextPixel = n5
- Rule #5: if n1=0, then nextPixel = n1
- Rule #6: else nextPixel = n3

There are four variations of drop algorithm: Top-Left (TL): applied on original image, Top-Right (TR): applied after flipping original image vertically, Bottom-Left (BL): applied after flipping original image horizontally and Bottom-Right (BR): applied

after flipping original image horizontally as well as vertically. Before applying drop-fall, the images are complemented as per requirement of the algorithm.

The Figure 12 shows the effects of drop-fall applied on flipped images and original image.

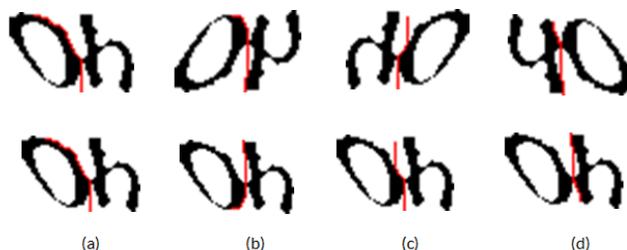


Fig 12. Drop-fall results a) Top-Left (as original) b) Top-Right c) Bottom-Left and d) Bottom-Right

3.3 Recognition

The recognition methods for text based CAPTCHAs are generally divided in three categories: template matching, character feature and machine learning. Amongst machine learning methods, artificial neural networks and deep learning to have produced better results. The artificial neural networks are also used for speech recognition besides text recognition but extracting character features explicitly are proved as a bottleneck on learning rate of neural network. These days deep learning has gained notable success in recognizing images, audio and text therefore convolutional neural network is used for recognition process. Convolutional Neural Network is a type of deep Neural Networks and most suitable for image recognition and image classification. The major advantage of CNNs over other traditional classification algorithms is its independence of human effort in feature design. In CNNs there are mainly two types of layers: convolution and pooling which passes through many activation layers for producing final output. Convolutional layers are used to transform input image for extracting features from it and during transformation, image is convolved with kernel. Pooling layers are used to reduce the dimensionality of images by reducing the number of pixels in the output of the previous convolutional layers. The output of these convolutions and pooling layers are fed into fully connected layer and the end result of fully connected layer helps in image classification depending on probabilities amongst classes. The proposed CNN model is shown in (Figure 13).

4 Results

Data was generated synthetically and then trained 3600 CAPTCHA images for each type (i.e. 3 letters, 4 letters and 5 letters CAPTCHA images). The 3600 CAPTCHA images for 5 letter created $5 * 3600 = 18000$ random images to be trained. 66% of images were trained and rest were kept for testing and validation. The generated CAPTCHA images were segmented twice. During initial segmentation, individual characters as well as connected components were obtained and then connected components were further segmented to obtain single characters. At this stage more than 90% letters were successfully segmented. The segmented characters were trained using CNN model. The results for 3, 4 and 5 letters of CAPTCHAs were recorded around 77.5%, 64.2% and 51.9% respectively. (as shown in Table 2). The overall precision of the method depends on the segmentation success rate, the recognition success rate of model and the number of letters in tested CAPTCHA⁽⁵⁾ as shown in the equation (1)

$$OP(\%) = SSR\% (SRR\%)^N \dots \dots \dots \text{Equation (1)}$$

Where

OP stands for Overall Precision,

SSR stands for Segmentation Success Rate and

SRR stands for Segmentation Recognition Rate

SSR depends on how many numbers of characters are correctly segmented in a dataset of images.

For example if a segmentation algorithm can segment 400 characters in 100 images of 5-Letter CAPTCHAs (100*4, as there are 5 characters in each image) then the segmentation accuracy would be calculated as $400/500 = 0.800$ or 80.0%. SRR depends on the accuracy of the classifier. We obtained satisfactory results. In the future, other neural networks with different combinations will be used.

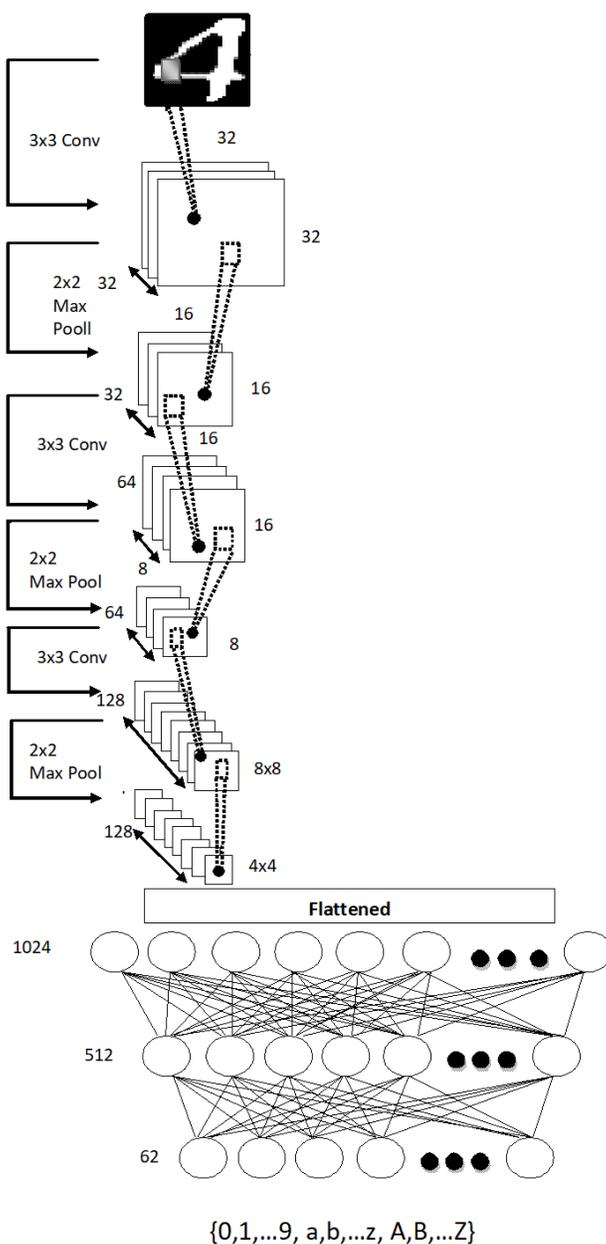


Fig 13. Convolutional Neural Network Model

Table 2. Recognition results

CAPTCHA	SSR (%)	SRR (%)	OP (%)
5 Letters	90.5	89.5	51.9
4 Letters	92.5	91.3	64.2
3 Letters	94.9	93.5	77.5

5 Conclusion

In this work, 3 variations of complex text-based CAPTCHA images were generated synthetically with varying number of letters, dimensions of images and distortions. A framework was proposed to decipher the complex text-based CAPTCHAs. The proposed framework based principally on conventional steps of decoding CAPTCHAs (i.e. preprocessing, segmentation and recognition). During preprocessing, simple image processing techniques: thresholding, binarization and erosion were applied to highlight the required text characters and to remove the noise. The mandatory segmentation was based on 4-connectivity to extract individual characters and connected components and the discretionary segmentation based on drop-fall algorithm to extract single characters from connected components. Deep neural network technique (i.e. CNN) was used for recognition and satisfactory overall precision of 77.5%, 64.2% and 51.9% were achieved for 3, 4 and 5 letter CAPTCHAs respectively.

In the Future, instead of extracting individual character and then recognizing single characters, a holistic approach involving variable length of CAPTCHA characters will be tested with different deep neural network schemes.

References

- 1) von Ahn L, Blum M, Langford J. Telling humans and computers apart automatically. *Communications of the ACM*. 2004;47(2):56–60. doi:10.1145/966389.966390.
- 2) Baird HS, Luk M. 2003.
- 3) Chellapilla K, Larson K, Simard PY, Czerwinski M. Computers beat Humans at Single Character Recognition in Reading based Human Interaction Proofs (HIPs). *InCEAS*. 2005.
- 4) Simard PY, Steinkraus D, Platt JC. Best practices for convolutional neural networks applied to visual document analysis. *InIcdar*. 2003;3.
- 5) Mori G, Malik J. Recognizing objects in adversarial clutter: Breaking a visual CAPTCHA. *In2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. 2003;1.
- 6) Yan J, Ahmad E, Low AS. 2008.
- 7) Chandavale AA, Sapkal A. A new approach towards segmentation for breaking CAPTCHA. *InInternational Conference on Security in Computer Networks and Distributed Systems*. Berlin, Heidelberg. Springer. 2012.
- 8) Starostenko O, Cruz-Perez C, Uceda-Ponga F, Alarcon-Aquino V. Breaking text-based CAPTCHAs with variable word and character orientation. Elsevier BV. 2015. Available from: <https://dx.doi.org/10.1016/j.patcog.2014.09.006>. doi:10.1016/j.patcog.2014.09.006.
- 9) Ahmad ASE, Yan J, Tayara M. The robustness of Google CAPTCHAs. *Computing Science*. Newcastle University. 2011.
- 10) Chellapilla K, Larson K, Simard P, Czerwinski M. Designing human friendly human interaction proofs (HIPs). *InProceedings of the SIGCHI conference on Human factors in computing systems*. 2005;p. 711–720.
- 11) Moy G, Jones N, Harkless C, Potter R. Distortion estimation techniques in solving visual CAPTCHAs. *InProceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. 2004;2.
- 12) Arain RH, Shaikh RA, Kumar K, Maitlo A, Kehar A, Shah SA, et al. Verifying the Robustness of Text-based CAPTCHAs offered by Local E-Commerce Sites. *INTERNATIONAL JOURNAL OF COMPUTER SCIENCE AND NETWORK SECURITY*. 2018;18(9):79–84.
- 13) Blumenstein M, Verma B, Basli H. A novel feature extraction technique for the recognition of segmented handwritten characters. *Seventh International Conference on Document Analysis and Recognition*. 2003;p. 137–141.
- 14) Cao Z, Huang M, Wang Y. A new drop-falling algorithms segmentation touching character. *In2010 IEEE International Conference on Software Engineering and Service Sciences*. 2010;p. 380–383.
- 15) Huang SY, Lee YK, Bell G, Ou ZH. A projection-based segmentation algorithm for breaking MSN and YAHOO CAPTCHAs. *InICSIE'08: Proceedings of the 2008 International Conference of Signal and Image Engineering*. 2008.
- 16) Hussain R, Gao H, Shaikh RA. Segmentation of connected characters in text-based CAPTCHAs for intelligent character recognition. *Multimedia Tools and Applications*. 2017;76(24):25547–25561. doi:10.1007/s11042-016-4151-2.
- 17) Karim S, Zhang Y, Laghari AA, Asif MR. Image processing based proposed drone for detecting and controlling street crimes. *Int Conf Commun Technol Proceedings, ICCT*. 2018;p. 1725–1730.
- 18) Shaikh RA, Memon I, Hussain R, Maitlo A, Shaikh H. A contemporary approach for object recognition based on spatial layout and low level features' integration. *Multimedia Tools and Applications*. 2018. doi:10.1007/s11042-018-6796-5.
- 19) Ibrar M, Mi J, Karim S, Laghari AA, Shaikh SM, Kumar V. Improvement of Large-Vehicle Detection and Monitoring on CPEC Route. *3D Research*. 2018;9(3). doi:10.1007/s13319-018-0196-5.
- 20) Laghari AA, He H, Shafiq M, Khan A. Assessment of quality of experience (QoE) of image compression in social cloud computing. *Multiaagent and Grid Systems*. 2018;14(2):125–143. doi:10.3233/mgs-180284.
- 21) Karim S, Zhang Y, Yin S, Laghari AA, Brohi AA. Impact of compressed and down-scaled training images on vehicle detection in remote sensing imagery. *Multimedia Tools and Applications*. 2019;78(22):32565–32583. doi:10.1007/s11042-019-08033-x.
- 22) Yan J, Ahmad E, Low AS. 2008.
- 23) Hussain R, Gao H, Shaikh RA, Kumar K. Recognition of text-based CAPTCHAs with neural confidence. *International Journal of Computer Science and Information Security*. 2016;14(9):290–290.
- 24) Shujra AA, Rajper S, Jumani AK. Measurement of E-learners' level of interest in online course using Support Vector Machine. *Indian Journal of Science and Technology*. 2019;12(40):1–9. doi:10.17485/ijst/2019/v12i40/147265.
- 25) Jumani AK, Mahar MH, Khoso FH, Memon MA. Online Text Categorization System Using Support Vector Machine. *Sindh University Research Journal -Science Series*. 2018;50(001):85–90. doi:10.26692/surj/2018.01.0014.
- 26) Jiangqing W, Wei C. Vertical Projection Characters Segmentation Based on Minimum Threshold and Curve-Fitting. *Journal of South-Central University for Nationalities (Natural Science Edition)*;2011(4):22–22.
- 27) Jumani AK, Memon MA, Khoso FH, Sanjrani AA, Soomro S. Named entity recognition system for Sindhi language. *InInternational conference for emerging technologies in computing*. Cham. Springer. 2018.

- 28) Siddiqui MF, Siddique WA, Ahmedh M, Jumani AK. Face Detection and Recognition System for Enhancing Security Measures Using Artificial Intelligence System. *Indian Journal of Science and Technology*. 2020;13(09):1057–64.