CAPTCHA Recognition using Deep Convolutional Neural Networks (DCNN)

Canavoy Narahari Sujatha,
Electronics and Communication engineering,
Sreenidhi institute of science and technology,
Hyderabad, India.
cnsujatha@gmail.com

Chilukoti Pranathi,
Electronics and communication engineering,
Sreenidhi institute of science and technology,
Hyderabad, India.
pranathichilukoti150@gmail.com

N. Hari Kumar reddy,
Electronics and communication engineering,
Sreenidhi institute of science and technology,
Hyderabad, India.
harikumarreddy85@gmail.com

Gandu Sushma,
Electronics and communication engineering,
Sreenidhi institute of science and technology,
Hyderabad, India.
gandusushma11@gmail.com

Abhishek Gudipalli,
School of Electrical Engineering,
Vellore Institute of Technology,
Vellore, India
abhishek.g@vit.ac.in

Abstract-This paper evaluates on captcha recognition using deep convolutional neural network. CAPTCHA is completely automated computer program which is used to publicly test and recognise human and robot separately. The captcha software builds very ambiguous pattern of alpha-numeric to protect the site being accessed by the robots, but these captchas maybe tough to crack by humans too. By using the deep learning model which can identify the captcha by studying the pattern of alphabets and numbers. This method can be further be used for handwriting recognition and number plate recognition. The major key feature in order to decipher the captcha was the pattern of the alphabet and the number, in the deep learning the convolution neural networks are the best to approach this. In convolutional neural network the feature maps help in identifying or recognizing the feature. To render the best outcome deep convolutional neural networks (DCNN) are used. As the deep CNN has many layers efficient and effective computation is done. A simple CNN can do the work but here the captcha has a very confusing forms which may not be effectively solved by the CNN alone. The convolutional operations performed on the sub matric of the picture of a captcha with the feature map matrix which has less index than the sub matric, the sub matric which accounts average convolutional value 1 is said to be the feature which is aimed to be recognized. As deep CNN has comparatively more layers it can compute the model deeply. This method helps the robots not identify the captchas and humans can use the deep learning models to solve it, and the robots are unaware of it. Hence, the DCNN method was the most efficient way to achieve the objective of CAPTCHA recognition with the required accuracy and minimal loss.

Keywords: CAPTCHA, Convolution, Deep CNN, Feature map, Handwriting recognition, Humans, Number plate recognition, Pattern recognition, Robots.

I. INTRODUCTION

To utilize this technology of captcha recognition for hand writing prediction and number plate identification the ultimate goal can be reached. Initially several websites moment still use traditional Captchas, a combination of Letters and figures. These websites end up being at threat of their data getting streamlined automatically by bots. The thing of this design isn't only to directly identify the Captchas but also to understand what characteristics a Captcha should have in order to be effective. The captcha recognition can be a very challenging security system to achieve in the present days to prevent the sites to be accessed by the unauthorised users. Sometimes it is a challenging task to provide a machine to produce a toughest captcha for the human users to crack, in the present day that is achieved but it has backfired as a toughest nut to crack for the human users. This project was thought to develop to solve such kind of issues as this technology is classified for any kind of robots. The initial trials were to find the best plausible technology to solve this problem, the technologies like convolutional methods were the accurate approach found, the convolutional kind of neural networks are used for the image segmentation and analysing them pixel wise can do the job for a particular extent, the accuracy was the challenge faced here. So, the accurate approach was to increase the number of intermediate layers in between. Which lead to the idea of deep convolutional neural network. In this neural network the number of layers increase and hence the computational model also increases and provide a proper accuracy. The model build has two convolutional blocks which is used for two kinds of images which has different kernel matrices, one for the 32 index and other for the index-64. The loss function used here is the CTC which again acts as another layer to calculate the loss, here the CTC stands for the connectionist temporal classification. These layers help in finding the loss which refers to the accuracy where the model is able to predict correctly and where the model is failing to predict the outcome correctly. The activation function is the softmax, this kind of activation function is used to operate at the output layer for classification purpose, the progress goes in such a way that at the very starting point the character set is fed to the model and then the input images are given to our deep CNN model where the segmentation and recognition of the characters in the captcha is recognised and for the proper prediction at the output layer the softmax activation function is applied to estimate the thresholds one recognised character is reaching and classifying it accordingly. LSTM is used for preventing the loss over a long period of time as it stands for long short-term memory. This keeps the model from losing its memory when it is used for the future purposes. A strong feedback network is also embedded in it using the recurrent neural network (RNN). This RNN is used for the impulse the reduction of errors then and there. The model gives the accuracy for about 90 percent and a bit of losses. This model can be considered for real time life application for various computational purposes. This approach has solved the problem statement. The development of captcha technology is seen in Figure 1 below, where the system or online web application sites originally require the user or the person utilising the site to read the unreadable style text to be typed on the designated space to be input. After that, the submitted text would be examined or compared to determine how closely it resembled the CAPTCHA, with a predicted similarity index %. If the similarity index reaches the necessary threshold, the user is considered to be human. If not, it's probable that the visitor is a robot, in which case the site's security features would bar visitors who aren't permitted from accessing it and would refuse the visitor access. This method was improved over time to further tighten security and create captchas that were more challenging. Additionally, we use image recognition technology to distinguish between human and robot users and to anticipate a robot's attempt to gain unauthorised access to a particular website. Users are tasked with finding a certain object in photographs that were probably decoded by human intelligence. As a result, captcha technology has progressed from recognising important words to identifying important images.

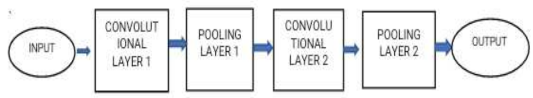


Figure 1. Block diagram of deep convolutional neural networks.

Figure 1 depicts a block diagram of the working of the deep convolutional neural networks. In this the deep indicates the number of layers we used like it may be 5 or 10 based on the problem statement. The main function of this convolutional neural networks is to find the patterns in the images and videos. In convolutional layer it takes the weights and multiplies them with the inputs from the neural network. The pooling layer gradually reduces the size of the pattern keeping only the required one.

The achievement for the captcha recognition can be done in various ways by seeking different deep learning methods like the autoencoders, Convolutional neural networks and deep convolutional networks. This project manifestation is done through the deep convolutional neural network approach. In this deep convolutional neural network, a couple more layers are added to allow the computational model for the character recognition to be more intense and efficient. By simple using the convolutional neural network the top-notch results couldn't be achieved so moved to the advanced method which id deep convolutional neural network shortly referred to as the DCNN. By using various deep learning techniques, such as deep convolutional networks, autoencoders, and convolutional neural networks, it is possible to perform captcha recognition in a variety of ways. Deep convolutional neural networks are used in this project's manifestation. A few extra layers are added to this deep convolutional neural network to make the character recognition computational model more powerful and effective. The best results couldn't be obtained by only utilising a convolutional neural network, so we switched to an advanced technique called a deep convolutional neural network, or DCNN for short.

For developing the captcha recognition technology with the help of deep convolutional neural networks many other approaches from the pre-existing methodologies are studied by referring to the journal based on the captcha recognition are studies and a brief insight of the observation from those journals are mentioned in the next session.

II. LITERATURE SURVEY

Jun Chen Proposed grounded CAPTCHA recognition using the being styles of pre-processing and segmentation- grounded styles. In this paper the authors have explained how different kinds of texts are recognised based on their colour, font style, text pattern, font size etc. They have followed mainly five processing of text in their model which are the segmentation, recognition, combination and pre-processing. The authors have seeked the best approach with the CNN. The book grounded CAPTCHA explains the different methods of breaking techniques, like how the text is segmented and analysed to predict what character it is and also depicted the advantages and disadvantages obtained by the aspects of understanding the captcha which are the five mentioned. These references are the used for studying the texture of each character. grounded on this reference with the help of convolutional neural networks (CNN) we progressed [1].

This paper, (Yu hu,2018) they have approached a very different solution in CNN by ignoring the existing conventional methods in the CNN such as the segmentation and character location. This reference was very useful to our project [2].

L. Von Ahn, proposed a unique CAPTCHA recognition model for differentiating the data typed in the computer and handwritten data of a human being which is different and random and varies for person to person and each time. This model can be a very close compared to the ideation of this paper, which was very useful to recognise human and a computer separately [3].

N. Roshanbin and J. Miller This paper, proposed a CAPTCHA recognition model that can interact with the mankind, the origin of this approach is that many technologies in CAPTCHA such as audio- centric, video-centric kind of technologies were approached to make the best out of it and make the intruders a hard task for attacking because there is no chance a robot can mimic human appearance or sometimes a human voice, but this approach caused them a storage problem for data as these files of audio and video are absolutely large in size. The text centric CAPTCHA recognition is not that accurate so interaction with human is the only way [4].

In this project the authors have took a challenge with the Microsoft's MSN CAPTCHA model, which has a segmentation expertise but it can be easily cheated by a very clever approaches by the attackers, so these authors have introduced a similar kind of model that has a 92 percentage of accuracy and no cleaver unauthorised attackers can find a way to break it. This idea has helped a lot with the segmentation part in this project which can be applied and made useful for the end goals [5].

S.-Y. Huang (2008) This paper has solved the challenges with the segmentation processing in the MNS captcha and yahoo captcha recognition. The main aim of the authors was to solve and improvise the services provided by these two companies. They had a relative comprehensive analysis in both the segmentation services provided by both the companies. In one of the solved approaches that had 18 percent of increase in the accuracy and in the other approach they had up to 32 percent of increase in the previous accuracy segmentation rate. This approach has given a head start for segmentation processes [6].

In this paper, First, they have taken forward the work of previously mentioned authors who have cited the recognition of the character styles in the handwriting and the CAPTCHA. The book was called the textbook- grounded CAPTCHA, similar to them they have introduced a variety of image text analysis methods which are very useful for this project. They also have proposed segmentation and grey-scaling approaches for the text images. Just like them they have also inculcated the five aspects where the image is being processed and recognised at the vey output [7].

In this paper, they explore the study they have performed on 197 scenarios where each number is assigned to a person and made to crack the captcha, they have chosen the person by the amount of internet experience they have and their age and experience. This study amounted to analysis sheet giving a valid information about how to build a efficient way of CPATCHA recognition model. [8].

The authors have cited the various methods of attacks the intruders use to access the sites unauthorised. They depict the different levels for attacking done by the hackers and provides the solutions for how to backfire them and to prevent them in different cases and was helpful to understand the CAPTCHA concepts [9].

In this paper, they have designed and tested a "rear" Turing test grounded on "pessimal print" and shown that it has the eventuality of offering a dependable and completely automatic system for telling people and machine druggies piecemeal over GUI interfaces. this exploration was a veritably good reference for our design [10].

In this paper they have implement different types of CAPTCHAs, but text-based schemes are the most widely used due to its easiness and robustness. [11].

In this paper that was referred for more information on – how segmentation is used for image identification was very useful [12] – [14].

In this paper A. S. El Ahmad and J. Yan, and L. Marshall have depicted a workshop conducted by them, this referred to robustness of the CAPTHA nowadays and introduced concepts to secure the CAPTCHA identification by the robots. [15].

III. METHODOLOGY OF PROPOSED WORK

This session involves the elaborating of the approach to proposed methodology which is deep convolutional neural network after analysing all the fit falls of other methods.

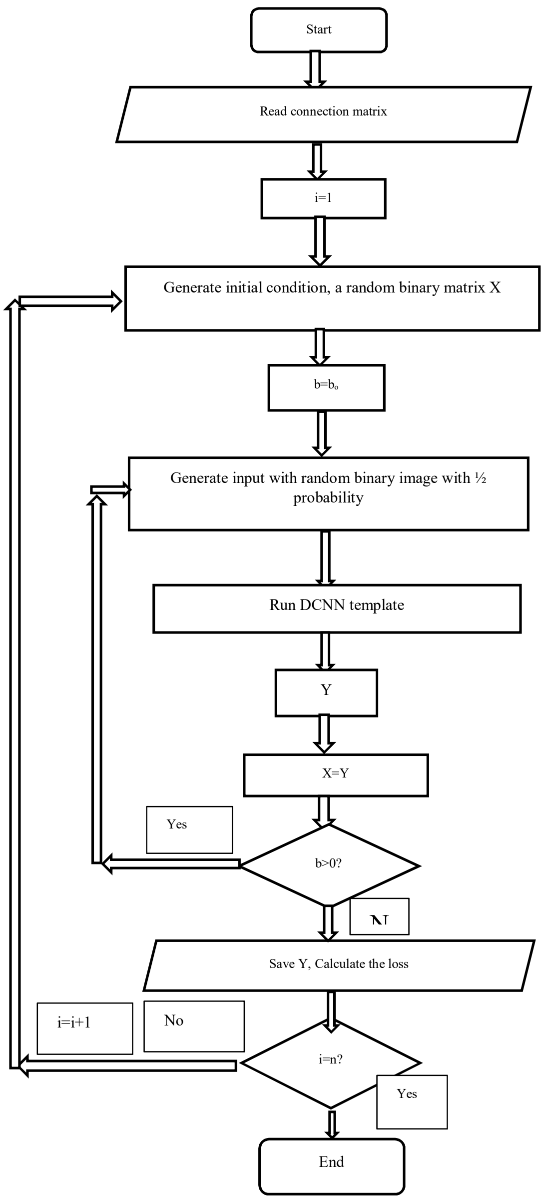


Figure 2. Algorithm for Deep convolution neural network

From analysing the methodologies followed by other authors which are Convolutional neural network (CNN), autoencoders and other supervised learning methods, the most suitable methodology found was the deep convolutional neural network (DCNN), though it is partly similar to the CNN model the DCNN brings out the most efficient outcome. CAPTCHA recognition using deep CNN is a kind of supervised learning where the alphabets and the numbers and their patterns are recognised and classified under a name where the similar pattern letters are recognised and grouped. Figure 2 shows the algorithm for the deep convolutional neural networks where the initialization of the model is done by reading the connected matrix (X) of the input image then it is further divided into the binary image matrix (Y) to perform the dot product in the convolutional layer then it is evaluated for satisfying the accuracy and loss function of the matrix Y. Till this condition is satisfies the algorithm runs into a loop.

The DCNN is found to be the most efficient because it consists of many layers with which is can compute the CNN model with much more deep layers or the multiple intermediate layers. The Algorithmic flow chart as shown in the figure 2 depicts the mathematical representation of the same process where it runs into a loop by means of a variable ‘i’ until it reaches the aimed accuracy and efficiency. Compared to the conventional CNN model the deep CNN has more deep layers to compute and the feature maps are even more vivid than the usual ones that are used in the conventional CNN.

The feature maps are made more complex and vivid and helps the model to train in such a way that it can realise or recognise a mere feature which means even a very little detail that is related to a known pattern. Sometimes the CAPTCHA may be too confusing due to its advancement and accuracy, its software will create more complex patterns for minute feature maps are a necessary in this case. Using a deep learning model that can recognise a captcha by looking at the arrangement of the letters and digits. Other applications for these techniques include number plate and handwriting recognition. The pattern of the letter and the number was a crucial component in cracking the captcha, and convolution neural networks in deep learning are the perfect tool for the job. The feature maps in convolutional neural networks aid in locating or recognising the feature. Deep convolutional neural networks (DCNN) are employed to produce the best results. Given that the deep CNN includes multiple layers, computation is done efficiently and effectively. A straightforward CNN can complete the task, however in this case the captcha has very complex shapes and might not be successfully solved by the CNN alone. Convolutional operations are carried out on a captcha image's sub matrix using a feature map matrix that has less indexes than the sub matrix; the sub matrix that accounts for the average convolutional value of 1 is referred to as the feature that is intended to be recognised. Deep CNN can calculate the model deeply because it contains substantially more layers. With the aid of deep learning models, humans can use this strategy to answer the captchas while keeping the robots in the dark. Hence, the DCNN approach was the most effective means of achieving. The implementation and the results obtained from the project by using this proposed work methodology will be shown in the results and discussion session. The dataset used and the splitting of training set and the validation set are discussed as well.

IV. RESULTS AND DISCUSSION

This session demonstrates the results obtained by using the above methodology, DCNN. In this architecture a total of fifteen layers are used, in which two different 2D convolutional layers are used, one for image of matrix size 64*64 and other is for the image of matrix of 32*32. For both the input images of two different matrix indices a kernel matrix of index 3*3 is used to apply as a filter or a feature map. These two convolution layers are computed under the connectionist temporal classification layer shortly known as CTC layer which is mostly used for training the recurrent neural network used in this approach later for minimizing the error using the feedback method. The dataset used in this is a .png files of a total 1040 different captcha images, are further divided into training set and validation set using keras library in dee learning. In the training set 936 images out of the given 1040 super dataset which is 90 percentage is used and for the validation process 104 images from the super dataset which is 10 percentage is used, this can be observed in the figure 5 and the data generated is pre-processed as shown in the figure 3 and 4 and while building each layer kernel initializer is used and ReLu activation function is applies at the prior of the computational layers. The platform used to implement this model is the Kaggle notebook. In this note book the implementation is done by importing all the required libraries such as the keras which is used to build a deep learning model, matlibplot to plot the pixel nodes in the 3D plane, computer vision to process and analyse the given images, OS module from python to ass some plugins, numpy for arranging the matrices to understand the images and pandas for working with the imported datasets. This platform is very open source and a web application which is very flexible to use and compute the deep learning models and mathematically analysed outputs are renders unlike any other platform. There are several steps followed to transverse the deep convolutional neural network. Data generation, data pre-processing, splitting the training and validation dataset, running epoch for a variable time, rendering the output and comparing the actual values to the rendered or predicted outcomes. The obtained outputs are shown in the following demonstration with the corresponding images.

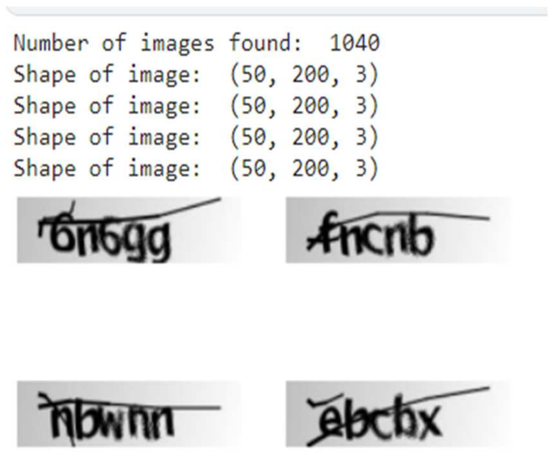
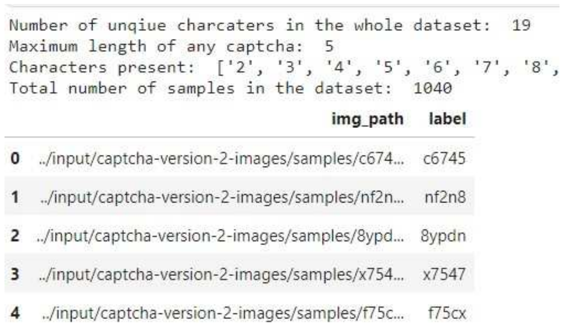


Figure 3. Analysis of data set

Figure 3 gives the Idea of the number of images the dataset is holding and also the geometrical information about the captcha images files in the form of a tuple as shown in the above figure, it shows the height and width as 50 and 200 pixels respectively and the axis of the image as 3. This is shown for the set of the image which has 4 images in it at a time.



```
Number of unqiue charcaters in the whole dataset:  19
Maximum length of any captcha:  5
Characters present:  ['2', '3', '4', '5', '6', '7', '8',
Total number of samples in the dataset:  1040
```

	img_path	label
0	../input/captcha-version-2-images/samples/c674...	c6745
1	../input/captcha-version-2-images/samples/nf2n...	nf2n8
2	../input/captcha-version-2-images/samples/8ypd...	8ypdn
3	../input/captcha-version-2-images/samples/x754...	x7547
4	../input/captcha-version-2-images/samples/f75c...	f75cx

Figure 4. Unique character set

Figure 4 shows the unique characters which includes alphabets and numbers, this helps the model to gain a basic knowledge on the characters which might look similar to those on the captcha. This set may also include the special characters for more knowledge base. But in the above figure we can see that the pre process depicts that maximum captcha length is assigned as 5 and the probable character set is mentioned in the flower braces.

```
Number of training samples:  936
Number of validation samples:  104
Number of training images:  (936, 50, 200)
Number of training labels:  (936,)
Number of validation images:  (104, 50, 200)
Number of validation labels:  (104,)
```

Figure 5. Splitting of dataset

Figure 5 shows the splitting of testing and validation dataset. The testing dataset consists of 936 images of captcha which is 90 percent of the total 1040 images and te validation dataset consist of 104 images which is of 10 percent of the major dataset of 1040.

After splitting the dataset, the training dataset is made to transverse different layers of the deep learning model as shown in the figure 6. These layers help in convolving and computing the image dataset and help increase the model intelligence each time. Then while validating the model, it happens to identify some captcha characters from all the intelligence it gained.

```
Model: "model"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_data (InputLayer)      [(None, 200, 50, 1)]      0
_________________________________________________________________
Conv1 (Conv2D)               (None, 200, 50, 32)       320
_________________________________________________________________
pool1 (MaxPooling2D)         (None, 100, 25, 32)       0
_________________________________________________________________
Conv2 (Conv2D)               (None, 100, 25, 64)       18496
_________________________________________________________________
pool2 (MaxPooling2D)         (None, 50, 12, 64)        0
_________________________________________________________________
reshape (Reshape)            (None, 50, 768)           0
_________________________________________________________________
dense1 (Dense)               (None, 50, 64)            49216
_________________________________________________________________
dropout (Dropout)            (None, 50, 64)            0
_________________________________________________________________
bidirectional (Bidirectional (None, 50, 256)           197632
_________________________________________________________________
bidirectional_1 (Bidirection (None, 50, 128)           164352
_________________________________________________________________
dense2 (Dense)               (None, 50, 20)            2580
=================================================================
Total params: 432,596
Trainable params: 432,596
Non-trainable params: 0
```

Figure 6. Layers of the DCNN

Figure 6 explains the number of layers present in the deep learning model built and also manifests the position at which they are connected and how they are connected and also the param are noted on the side. As shown in the figure it gives info about the type of layer and the shape of the output obtained at the end of that particular layer.

The CTC layers which are known for their temporal classification nature has the factor of time effecting the output of the model. This temporal word refers to the time constrain for training the model for different images from the appropriate set. This demonstration can be seen in the below mentioned figure 7. Then at the output the accuracy is observed by comparing the ground truth which refers to a accuracy point for reference and the predicted outcome by the deep learning model under DCNN is shown in the figure 8.

```
Epoch 1/50
59/59 [==============================] - 15s 253ms/step - loss: 23.3809 - val_loss: 16.4328
Epoch 2/50
59/59 [==============================] - 13s 216ms/step - loss: 16.4130 - val_loss: 16.3877
Epoch 3/50
59/59 [==============================] - 13s 213ms/step - loss: 16.3879 - val_loss: 16.4333
Epoch 4/50
59/59 [==============================] - 13s 218ms/step - loss: 16.3778 - val_loss: 16.3301
Epoch 5/50
59/59 [==============================] - 13s 215ms/step - loss: 16.2529 - val_loss: 16.0426
Epoch 6/50
59/59 [==============================] - 13s 218ms/step - loss: 16.0566 - val_loss: 16.0255
Epoch 7/50
59/59 [==============================] - 13s 216ms/step - loss: 16.0026 - val_loss: 15.9814
```

Figure 7. Epoch run for 50 times

Figure 7 shows the epoch for training the model with the training dataset for 50 different times and the corresponding losses are shown accordingly and also the value of the loss is also shown beside it. This epoch can be run for variable number of times to have a visual representation for how the model is getting trained at each step in the neural networks.

The epoch is the function for a time dependent factor which can also be used for the computational training of the deep learning models such as the deep convolutional model which has the scope to show the accuracy and the loss at each step which means at each run of the training dataset. The epoch is an assigned value that takes the value to the maximum limit of the training dataset. As the training dataset consists of 936 images the manifestation of the epoch is limited to 936 times.

But it is a time taking process for such a large number so a limited value of 50 is taken to run the epoch and observe the accuracy and loss. The next step is the output where the outcome of the computed data is obtained after the pooling layer. This pooling layer is a very crucial layer for the computation at the last step.

Figure 8 shows the comparison between the actual and the predicted value. The reference is clear that the ground truth image code matches with the predicted outcome image code. This helps in identifying the accuracy of the model, the ground truth and the predicted values are shown for the 16 images of the CAPTCHA which is given to the model for training purpose, each and every value is tallied and checked, some model can have precision and other models have less precision according to the method of training, this model is at its at most accuracy which is seen below by observing the figure 8.

```
Ground truth: xw465      Predicted: xw465
Ground truth: 3n7mx      Predicted: 3n7mx
Ground truth: 5mnpd      Predicted: 5mnpd
Ground truth: edwny      Predicted: edwny
Ground truth: f364x      Predicted: f364x
Ground truth: f228n      Predicted: f228n
Ground truth: y7mnm      Predicted: y7mnm
Ground truth: xxw44      Predicted: xxw44
Ground truth: wnmyn      Predicted: wnmyn
Ground truth: ncyx8      Predicted: ncyx8
Ground truth: 728n8      Predicted: 728n8
Ground truth: mmy5n      Predicted: mmy5n
Ground truth: x347n      Predicted: x347n
Ground truth: ddcne      Predicted: ddcne
Ground truth: d4n82      Predicted: d4n82
Ground truth: p57fn      Predicted: p57fn
```

Figure 8. Comparing ground truth with predicted value

V. CONCLUSION

The proposed methodology concludes that the goal of achieving the captcha recognition can be done by a conventional convolutional neural network but the accuracy and the loss can be monitors and regulated by observing the results from much more advanced architectural computation which is deep convolutional neural network with multiple layers deducing the characters of the captcha. The deep convolutional neural network can be even used to extend it to other problem statements such as the hand writing recognition and number plate identification. After examining all the fit falls of existing approaches, the proposed methodology—deep convolutional neural network—is elaborated in this session. Convolutional neural network (CNN), autoencoders, and other supervised learning techniques were examined to determine the procedures used by other writers. The deep convolutional neural network (DCNN), albeit somewhat identical to the CNN model, produced the most effective results. Because to its several layers, which allow it to construct CNN models with numerous deep layers or intermediate layers, the DCNN is proven to be the most efficient. Figure 4's algorithmic flow chart represents the same process mathematically, running into a loop with the help of a variable called I until it achieves the desired accuracy and efficiency.

The deep CNN model has more deep layers to calculate than the standard CNN model does, and its feature maps are even more colourful than those employed in the conventional CNN. The model may train such that it can realise or recognise a simple feature, which implies even a very little detail that is associated to a recognised pattern. The feature maps are made more intricate and colourful. Due to the CAPTCHA's progress and precision, it can occasionally be too perplexing. In these circumstances, increasingly complicated patterns will be generated by the programme utilising minute feature maps. use a deep learning model that can identify a captcha by examining how the letters and numbers are arranged. Number plate and handwriting recognition are two further uses for these technologies. Convolution neural networks in deep learning are the ideal tool for the job since the letter and number pattern was a key factor in deciphering the captcha. Convolutional neural networks use feature maps to help find or identify features. To achieve the best results, deep convolutional neural networks (DCNN) are used. The deep CNN has numerous layers, which allows for efficient and effective computing. A simple CNN can do the task, however the captcha in this case has highly complicated forms and may not be solved by the CNN alone. Using a feature map matrix with fewer indices than the sub matrix of a captcha picture, convolutional operations are performed; The feature that is meant to be recognised is the sub matrix that accounts for the average convolutional value of 1. Because Deep CNN has a significant number of additional layers, it can compute the model in great detail. Humans may utilise this technique to solve the captchas while keeping the robots in the dark with the help of deep learning models. Thus, the DCNN method was the most efficient way to do.

References

1. Jun Chen, Yanqing Guo, Yi Zhang (2017), "A Survey on the breaking technique of Text-Based CAPTCHA Recognition", 15 pages, pp. 001-015, Dec.2017.

2. Yu hu, Li Chen and Jun cheng (2018), "A CAPTCHA Recognition Technology based on deep learning", Pp. 2158-2297, Jun. 2018.

3. L. Von Ahn, M. Blum, and J. Langford (2004), "Telling humans and computers apart automatically, Communications of the ACM", vol. 47, no. 2, pp. 56–60, Feb. 2018.

4. N. Roshanbin and J. Miller (2013), "A survey and analysis of current CAPTCHA approaches", Journal of Web Engineering, vol. 12, no. 1-2, pp. 001–040, Feb. 2013.

5. J. Yan and A. S. E. Ahmad (2008), "A low-cost attack on a Microsoft CAPTCHA, in Proceedings of the 15th ACM conference on Computer and Communications Security", CCS'08, pp. 543–554, Oct. 2008.

6. S.-Y. Huang, Y.-K. Lee, G. Bell, and Z.-H. Ou (2009), "An efficient segmentation algorithm for CAPTCHAs with line cluttering and character warping", vol. 48, pp. 267–289, Aug. 2009.

7. R. A. Nachar, E. Inaty, P. J. Bonnin, and Y. Alayli, (2015), "Breaking down Captcha using edge corners and fuzzy logic segmentation/recognition technique", pp. 3995–4012, Dec. 2015.

8. J. Tam, J. Simsa, and L. Von Ahn (2008), "Breaking audio CAPTCHAs, in Proceedings of the 22nd Annual Conference on Neural Information Processing Systems", NIPS 2008, vol. 21, pp. 1625– 1632, Jan. 2008.

9. A. L. Coates, H. S. Baird, and R. J. Fateman,(2001), "Pessimal print: A reverse turing test, in Proceedings of the 6th International Conference on Document Analysis and Recognition", pp. 1154–1158, Sep. 2001.

10. P. Golle (2008), "Machine learning attacks against the sierra CAPTCHA, in Proceedings of the 15th ACM conference on Computer and Communications Security", CCS'08, pp. 535–542, Oct. 2008.

11. P. Y. Simard, R. Szeliski, J. Benaloh, J. Couvreur, and I. Calinov, (2003), "Using character recognition and segmentation to tell a computer from humans, in Proceedings of the 7th International Conference on Document Analysis and Recognition", pp. 418– 423, Aug. 2003.

12. C. N. Sujatha and Y. V. Raghava Rao, "Video Steganography using LSB Scheme for Secure and Efficient Data Transmission", International Journal of Resent Technology and Engineering (IJRTE), Scopus Indexed, Vol. 7, Issue. 6S, Mar. 2019, pp. 724-727, Apr. 2019.

13. K. Chellapilla, K. Larson, P. Y. Simard, and M. Czerwinski, (2005), "Building segmentation-based human-friendly human interaction proofs (HIPs), in Proceedings of the Second International Workshop on Human Interactive Proofs", HIP 2005, pp. 1-26, May. 2005.

14. Parvathaneni Naga Srinivasu, Jana Shafi, T Balamurali Krishna, Canavoy Narahari Sujatha, S Phani Praveen and Muhammad Fazal Ijaz, "Using Recurrent Neural Networks for Predicting Type-2 Diabetes from Genomic and Tabular Data", Diagnostics 2022, 12(12), 3067, pp. 001-005, Dec. 2022.

15. A. S. El Ahmad, J. Yan, and L. Marshall, (2010), "The robustness of a new CAPTCHA," in Proceedings of the 3rd European Workshop on System Security", EUROSEC'10, pp. 36–41, May. 2010.