As per UGC guidelines an electronic bar code is provided to seure your paper

International Journal for Modern Trends in Science and Technology, 8(05):69-77, 2022

ISSN: 2455-3778 online
DOI: https://doi.org/10.46501/IJMTST0805011
Available online at: http://www.ijmtst.com/vol8issue05.html

CAPTCHAs for Improved Accessibility

Yashwant Kumar Jha[1] | Rishabh Kumar Singh[1] | Shallu Bashambu[2] | Ajay kaushik[2]

[1]Department of IT, Maharaja Agrasen Institute of Technology, Delhi, India
[2]Assistant Professor, Department of IT, Maharaja Agrasen Institute of Technology, Delhi, India

To Cite this Article
Yashwant Kumar Jha, Rishabh Kumar Singh, Shallu Bashambu and Ajay kaushik. CAPTCHAs for Improved Accessibility. International Journal for Modern Trends in Science and Technology 2022, 8(05), pp. 69-77. https://doi.org/10.46501/IJMTST0805011

Article Info
Received: 29 March 2022; Accepted: 27 April 2022; Published: 01 May 2022.

ABSTRACT

The internet has been playing an increasingly important role in our daily life, with the availability of many web services such as email and search engines. However, these are often threatened by attacks from computer programs suchas bots. To address this problem, CAPTCHA (Completely Automated Public Turing Test to Tell Computers and Humans Apart) was developed to distinguish between computer programs and human users. Although this mechanism offers good security and limits automatic registration to web services, but many CAPTCHAs are not user friendly and sometimes can pose some challenge to the users, especially visually impaired users. This paper provides an alternative OCR based captcha that can be used by all the users (including visually impaired users). It would provide better user experience, easy to solve and hard for a bot to crack.

KEYWORDS: CAPTCHA, Convolutional Neural Network, Handwritten Digit Recognition, Web Accessibility, Sliding Puzzle, Hill Climbing, Web Security

1. INTRODUCTION

CAPTCHA ("Completely Automated Public Turing test to tell Computers and Humans Apart") is a type of challenge-response test used by many web applications to ensure that the response is not generated by a computer. Visual CAPTCHA implementations are often vulnerable to various kinds of attacks.

Traditional captchas involve finding text from the given distorted image, or finding a particular object in given images. They are super useful in determining the difference between human and bots, but they are usually poor in user experience and are difficult for visually impaired users.

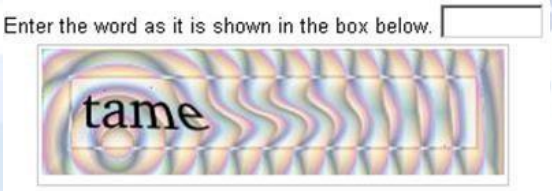


Figure 1.1: Text Based CAPTCHA

The main objective of the paper is to make the captcha verification process easy for visually impaired people. Traditional CAPTCHAs provide distorted or overlapping letters and numbers that a user then has to submit via a form field. Some latest captchas involve rotating an image or finding a particular object in given set of images. These tasks seem to be trivial for a normal user but they can pose some difficulties to visually impaired users. Users having weak eyesight can find it difficult to locate the text in given image. Color blind users cannot locate the desired object in the set of images.

The scope of this paper is to provide an alternative captcha system that can be used by visually impaired user as well. This captcha verification process will have a better user experience. Idea of is to provide the user some tasks that are easy for a human to perform but are difficult/impossible for a bot.

There are couple of options that we will give users to solve Captchas:

1. *OCR Captcha*

Users will be asked some basic questions like:

- What is the middle letter in 'INDIA'?
- Draw the smaller number: 3 or 9.
- What is 4-3?
- Draw letter that comes after 'D'

User will answer these questions by drawing their response on the screen. They will be provided a big canvas for drawing their response, and these questions will be read out for them using text reader. Their responses will be recorded and analyzed in the backend using Deep Learning algorithm. If the response is correct, then captcha verification is successful, otherwisethey will be asked some other random question.

In the backend, response of the user will be sent to Convolutional Neural Network (CNN) trained on handwritten characters dataset. After image preprocessing, CNN will give the results with high accuracy.

2. *Checkboard Captcha*

- A grid/table full of black and white blocks is to be displayed.
- User should be able to click on blocks to toggle the color between black and white.
- When user clicks submit button, check if all blocks are white.

3. *Spotlight CAPTCHA*

- A hidden CAPTCHA code (covered with black layer) is to be displayed.
- User should be able to hover the mouse pointer over the code to reveal the area below. Allow User to enter code in the input field.
- When user clicks submit button, check if what user typed matches with the code displayed

4. *Sliding Puzzle CAPTCHA*

- A sliding puzzle is to be displayed.
- When user clicks submit button, check if the Sliding Puzzle is in solved state.

o *Convolutional Neural Network*

In recent years, deep learning-based techniques have been gaining significant interest in the research community for solving a variety of supervised, unsupervised and reinforcement learning problems. One of the most well-known and widely used techniques are convolutional neural networks (CNNs), a kind of neural networks which are able to automatically extract relevant features from input data. The properties of convolutional neural networks, including the fact that they are able to retrieve features from multidimensional inputs, turn them into a very interesting alternative for solving problems within the field of computer vision. In fact, computer vision has become a testbed for validating novel techniques and innovations in CNNs.

More specifically, one dataset has been widely usedfor this purpose: the MNIST dataset. MNIST is a database of labeled handwritten digits, with separate training and test sets, and therefore is an easily interpretable domain that allows a fast comparison between different techniques. In fact, MNIST is so widely used that it is even used in "hello-world" tutorials of some deep learning frameworks, such as TensorFlow.

o *Breaking Traditional Captchas*

As per numerous reports [1], fraudsters find sophisticated ways to get past Captchas. For example, some bots that practice Captcha-solving depend on the artificial intelligence of the auto image or sound recognition. But another tactic, which is becoming more and more common, is relying on a time-tested problem- solving device i.e. the human brain. Surprisingly, Captcha farms are not about people assisted by bots but about human- assisted bots. Such huge database of solved generic CAPTCHAs is then used by spam bots. There are plenty of job postings online for captcha solving that pay people for bulk solving. This practice is majorly reported in developing countries where peopleare ready to solve at cheaper rates.

Fig 1.2: *CAPCTHA solving jobs*

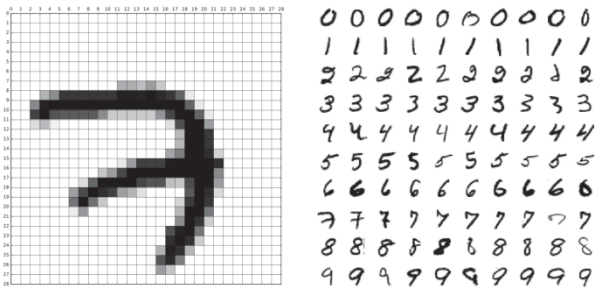
Greg Mori and Jitendra Malik of UC Berkeley Computer Vision Group, Simon Fraser University, in their research paper 'Breaking a Visual CAPTCHA' [2], write about how their method can successfully pass Yahoo's 'gimpy' CAPTCHA test 92% of the time using object recognition algorithm. Gimpy is a more difficult variant of a word-based CAPTCHA that generates a picture of distorted and skewed alphanumeric string. As a challenge the user is supposed to recognize the word embedded in the picture. Such vulnerability of generic CAPTCHAs has erupted a demand for better alternatives that are based on user interactions rather than mere text recognition.

2. METHODOLOGY

1. OCR CAPTCHAs

- **Data Collection:**

The dataset used in MNIST and EMNIST dataset. The MNIST database of handwritten digits contains 60,000 training examples and 10,000 testing examples, which are 28 * 28 images. And EMNIST is dataset of 320,000 images of handwritten alphabets. All the images have already been size- normalized and preprocessed and formatted (LeCun et al., 1998). In the process of loading the data set can be directly called from the MNIST database, but due to requirements of the assignment, I downloaded these image files from the website that provides the data set. Because downloading browsers may unzip these image collection files without your attention, this operation may cause the downloaded files to be larger than previously mentioned. Thus, if you need to see some problems with the original image set or data set, you can view the original site of the data set via the link provided in the reference section of the paper. Due to the use of Python's own data set, simplifying the section on data preprocessing in the code. The images are all centered in 28 * 28 field



(a) MNIST sample belonging to the digit '7'. (b) 100 samples from the MNIST training set.

Figure 1. Example of the MNIST database.

Convolutional Neural Networks

Due to the selection of the data set, we decompose the picture into 28*28 blocks of the same size. According to the original trained neural network, we input a complete picture into the neural network. But for CNN, the pixel block is directly input this time. The same neural network weight will be used for every small tile. If any small tile has any abnormality, we think the tile is interested. In this neural network, there is no order in which small tiles are disturbed, and the results are still saved in the order of input. Then we will get a sequence. The part where the picture is stored is interesting. Since the array is generally large, we will first down sample it to reduce the size of the array. Find the max value in each grid square in our array. Finally, the column will be inputted into the Fully Connected Network and the neural network will determine if the picture matches.

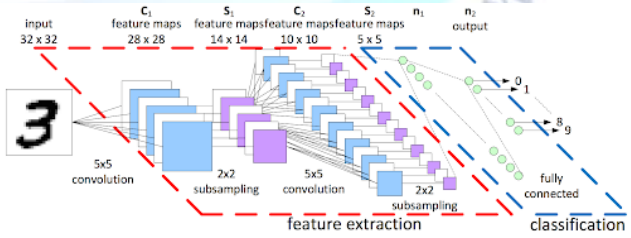


Figure 2: Convolutional Neural Network working.

- **Web-application Deployment**

Flask micro-framework would be used to make to web application, for easy integration with Convolutional Neural Network in the backend. Flask Rest API would act as a bridge between frontend and the backend. User will be given some basic mathematical questions and they will draw their response on the screen (canvas).

Their response will be sent to the backend, analyzed by the deep learning model and the result will be sent back.

This CNN trained model is deployed along with other types of captchas model that we have proposed. Users are given choice to choose the Captcha model in a dropdown menu while filling the form. Based on their response, respective captcha model will be displayed to the user. After successfully solving the Captcha, form will be submitted automatically.

Sliding Puzzle

For the working project used for observations, Hill climbing algorithm has been used which will be briefly introduced along with the slight variation adopted to design the solver-bot.

o **Hill Climbing Algorithm (with depth variation):**

Hill Climbing is a heuristic search used for mathematical optimization problems in the field of ArtificialIntelligence [4]. Given a large set of inputs and a good heuristic function, it tries to find a sufficiently good solution to the problem. This solution may not be the global optimal maximum.

- 'Heuristic search' means that this search algorithm may not find the optimal solution to the problem. However, it will give a good solution in reasonable time.
- A heuristic function is a function that will rank all the possible alternatives at any branching step in search algorithm based on the available information. It helps the algorithm to select the best route out of possible routes.

Features of Hill Climbing:

A. Variant of generate and test algorithm:

It is a variant of generate and test algorithm. The generate and test algorithm is as follows:

i. Generate possible solutions.
ii. Test to see if this is the expected solution.
iii. If the solution has been found, quit; else go tostep(i).

B. Uses the Greedy approach:

At any point in state space, the search moves in that direction only which optimizes the cost of function with the hope of finding the optimal solution at the end.

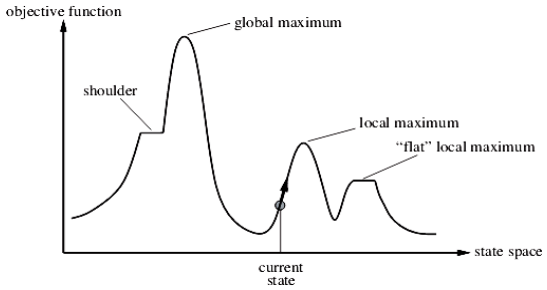


Fig 2.3: *Various regions in Hill-Climbing algo*

Hill climbing is an iterative heuristic-based search algorithm and is widely used in optimisation of search problems. In 'An Introduction to Machine Learning' book by Miroslav Kuba [5], it is defined as follows:

i. Create two lists, L and Lseen. At the beginning, L contains only the initial state, and L Lseen is empty.
ii. Let n be the first element of L. Compare this state with the final state. If they are identical, stop with success.
iii. Apply to n all available search operators, thus obtaining a set of new states. Discard those states that already exist in Lseen. As for the rest sort them by the evaluation function and placethem at the front of L.
iv. Transfer n from L into the list Lseen, of the statesthat have been investigated.
v. If L = 0, stop and report failure. Otherwise, go to ii.

Evaluation function at step iii basically just calculates the distance of the current state from the final state (Manhattan Distance [6]). For example:

Current state

1	2	3
	4	6
7	5	8

Goal state

1	2	3
4	5	6
7	8	

In the current state all values except (4,5 and 8) are at their respective places, so, the heuristic value is 3. (Total of three values are misplaced to reach the goal by distance of 1 each)

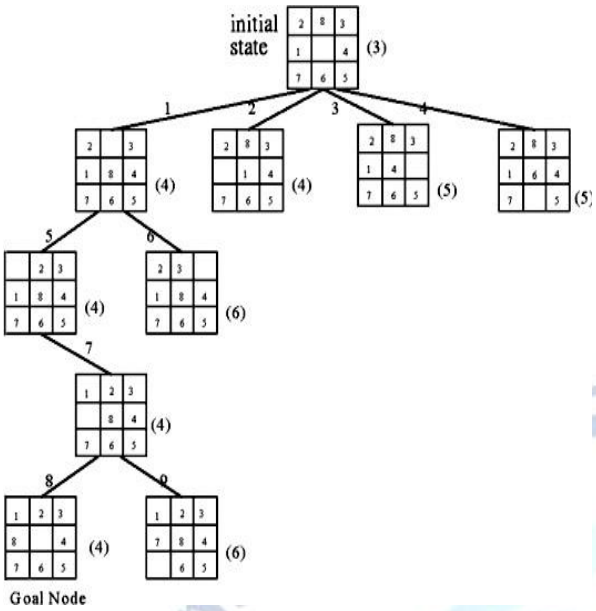


Fig 2.4: Sliding Puzzle state space (while usingHill-Climbing

- **Solving Sliding Puzzle with Hill- Climbing (with variable Depth):**

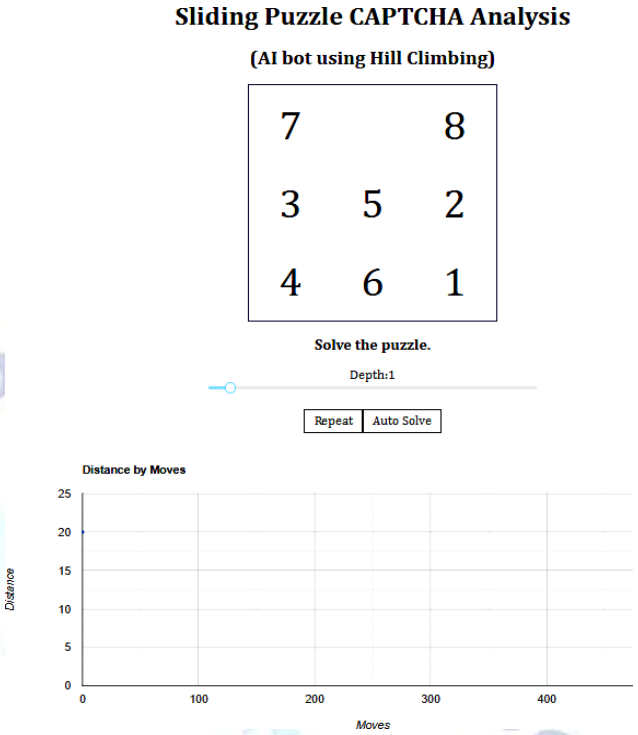
Instead of BFS or other non-heuristic searches, that are slower, the puzzle solver bot tries to solve the problem with Heuristic Search that is also known as Informed Search (Hill Climbing/A*)

To solve the problem with Heuristic search or informed search we have to calculate Heuristic values of each node to calculate cost function as discussed in previous part.

Moreover, to effectively chose the right path, Hill Climbing with the depth- first approach is put into use. The algorithm's idea revolves around traversal of a path for a defined number of steps(depth) to confirm that it's the best move.

i. Iterate over all the possible next moves/states for the current state.
ii. Redo step i until depth 'd' has been reached andthus generating a tree of height d.
iii. Pick the move/state with minimum cost(dF)

Return dF to parent node so that evaluation can be done at depth-1 level.



3. DESIGN AND IMPLEMENTATION

- **Image Augmentation:**

Image augmentation has been applied to the images to increase the size of dataset which ultimately increase the overall accuracy of the model during training.

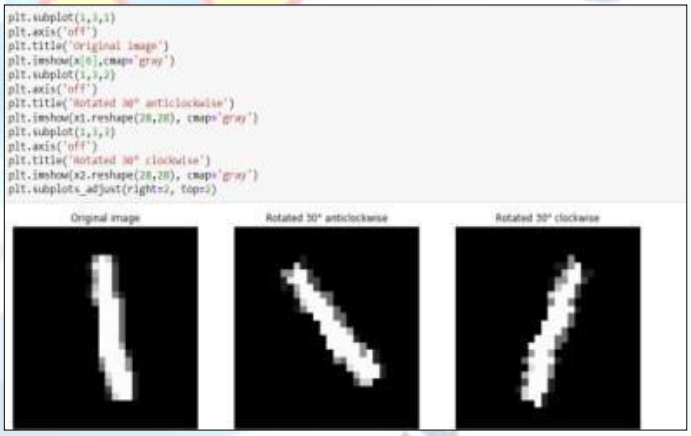


***Figure 3.1:** Image augmentation of dataset*

Convolutional Neural Network using TensorFlow

3-layered deep Convolutional Neural Network is build using TensorFlow library. Dataset was split into three parts: Training dataset, Testing dataset and Validation dataset. CNN was trained on training dataset, and tested on validation dataset to measure the training process and prevent the overfitting on model,

Model is then tested on testing dataset to know how better it will perform on some unseen dataset.

Training this CNN model on MNIST dataset took about 20-25 minutes when trained on GPU and accuracy above 99% was achieved.

To prevent overfitting, training was stopped when validation accuracy starts decreasing

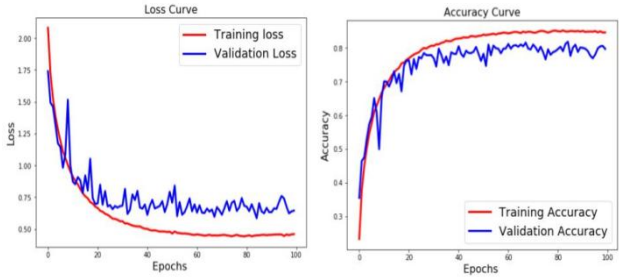


Figure 3: Convolutional Neural Network training

Image Preprocessing

The following image preprocessing would be carried out for each and every input image. First, the background of image is converted into black, and then proper cropping technique is applied to it. This will result in obtaining high accuracy on real world data

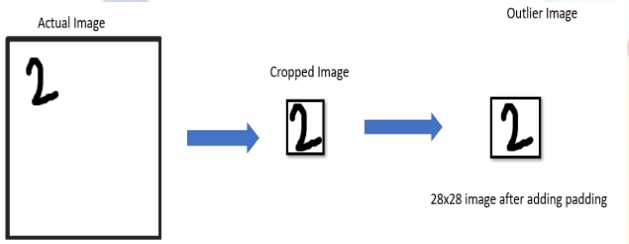


Figure 3.3: *Image preprocessing in webapp*

A.Proposed Workflow

Idea of is to provide the user some tasks thatare easy for a human to perform but are difficult/impossible for a bot. Users will be asked some basic questions like:

- What is 3+5?
- Draw the bigger number: 3 or 9?

Draw the middle letter in 'RIVER'

Draw last alphabet of English language

User will answer these questions by drawing their response on the screen. They will be provided a big canvas for drawing their response, and these questions will be read out for them using text reader. Their responses will be recorded and analyzed in the backend using the above trained Deep Learning algorithm. If the response is correct, then captcha verification is successful, otherwise they will be asked some other random question.

In the backend, response of the user will be sent to Convolutional Neural Network (CNN) trained on handwritten characters dataset. After image preprocessing, CNN will give the results with high accuracy.

In order for a bot to crack this captcha verification process, first it has to understand the question given to the user. Then have to find the answer of the given question. If the correct answer is obtained by the bot, it has to draw its answer on the canvas, which is a next level challenge. Also brute-force approach will not work in order to find the answer of the given question, because the questions are generated randomly.

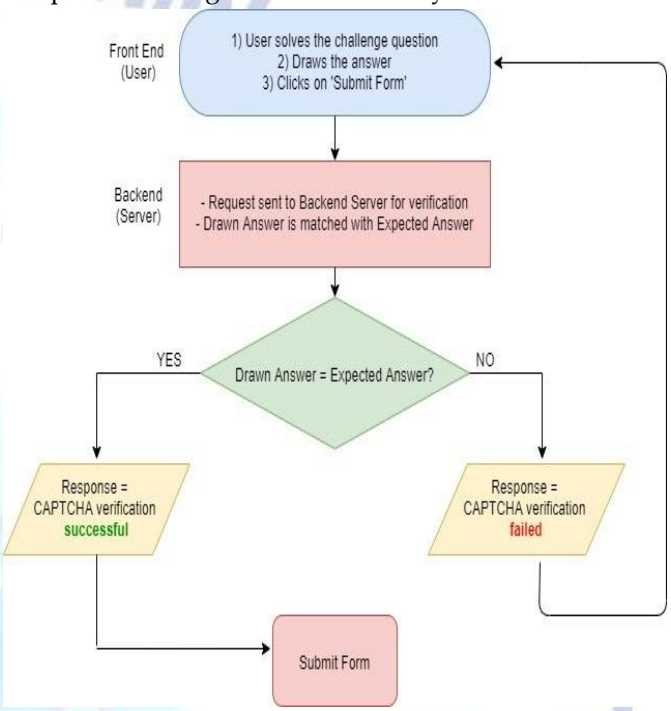


Figure 3.4: *Proposed workflow*

4. RESULTS

Training of Convolutional Neural Network

According to the model given above, the final accuracy is stable at more than 99%. In fact, for many tasks, the model is prone to over-training. If over training occurs during training, the error caused by training will decrease with the increase of training times. The following figure shows the accuracy of my model when the epoch is 12 times. Training set size also affects the accuracy, and accuracy increases as the amount of data increases. As shown in the figure below, the more data, the more data in the training set, the smaller the impact of training error and test error, and ultimately the accuracy can be improved. We can find the larger training set can develop the performance of LeNet- 5. After improving this problem, our accuracy can reach 99%. If we increase the number of training, this accuracy will be improved to some extent. But even if you increase the training set, you

will find that there is no way to achieve 100% accuracy. I attributed this part to the fact that some hand-written digits are too illegible. For this problem, we can understand that even if people recognize symbols and things on some pictures, they will encounter some patterns that are difficult to judge their specific meanings.

```
Epoch : 1 | Training Accuracy : 75.996% | Validate Accuracy : 96.695%
Epoch : 2 | Training Accuracy : 96.405% | Validate Accuracy : 97.460%
Epoch : 3 | Training Accuracy : 97.568% | Validate Accuracy : 98.003%
Epoch : 4 | Training Accuracy : 98.134% | Validate Accuracy : 98.352%
Epoch : 5 | Training Accuracy : 98.545% | Validate Accuracy : 98.568%
Epoch : 6 | Training Accuracy : 98.825% | Validate Accuracy : 98.670%
Epoch : 7 | Training Accuracy : 99.004% | Validate Accuracy : 98.756%
Epoch : 8 | Training Accuracy : 99.116% | Validate Accuracy : 99.283%
Epoch : 9 | Training Accuracy : 99.205% | Validate Accuracy : 99.389%
Epoch : 10 | Training Accuracy : 99.386% | Validate Accuracy : 99.566%
Epoch : 11 | Training Accuracy : 99.507% | Validate Accuracy : 99.567%
Epoch : 12 | Training Accuracy : 99.542% | Validate Accuracy : 99.130%
Accuracy on test dataset : 99.357
```

Figure 4.1: *Convolutional Neural Network training result*

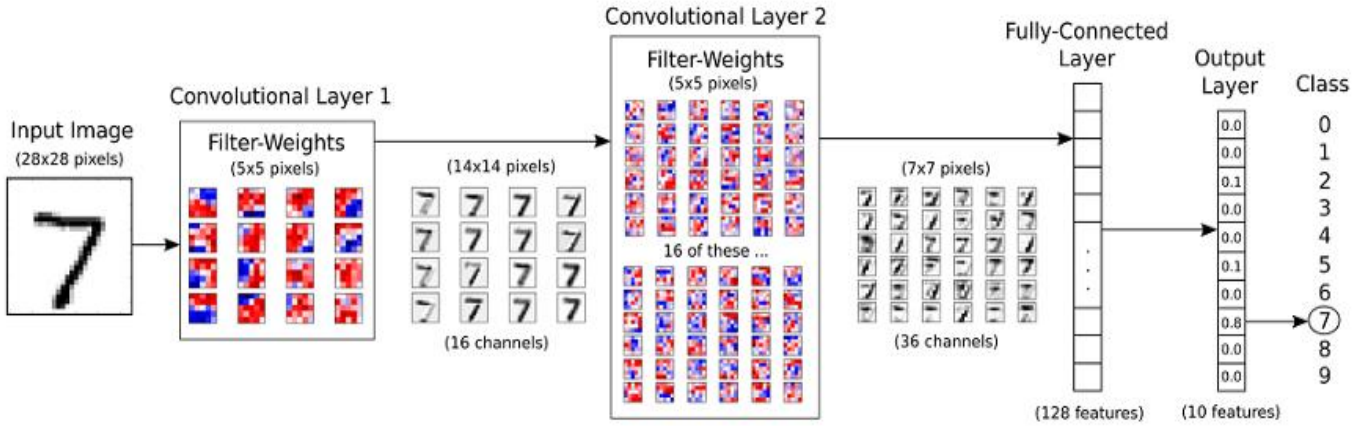


Figure 4.2: *Working of Convolutional Neural Network*

Web Application Result

User will be given choice to choose the captcha they want. There are total four captcha options available for user.

1. OCR Captcha

- User will be asked simple mathematical and alphabetical questions.
- These questions will be generated randomly, and user will answer them by drawing the digit/letter on screen.
- User can either submit their response or try again by clearing the canvas. Also, if the captcha verification is failed, user will be asked again a new question.
- For a bot to crack this, it should be able to draw on canvas. It is very difficult for a bot to understand the context of the question and then draw this on screen.
- When user draw their response on screen, it will be stored as an image in the backend and then analyzed by the deep learning algorithm (CNN).

If the response is correct, then captcha has been verified successfully verified.

2. *Black & White Checkerboard*

(User has to click all Black blocks in the grid until it is all-white.)

- Simple Html table has been used to create the grid for the checkerboard.
- Javascript has been used to randomly assign colour to the table cells (blocks).
- Mouse clicks on cell toggle their css attributes (mainly colour).

• An event listener is run to listen to any "click" event on the submit button to check whether there is any black block left.
• As a security measure, no. of clicks is also counted and sent to backend for verification.

On successful solving of challenge, a webpage redirection is made using jquery to the Success page

3. Spotlight CAPTCHA

(User has to hover mouse over the CAPTCHA code to reveal the area below.)

• A random code is generated using JavaScript.
• This code is embedded into html canvas to be converted from text to canvas image.
• CSS is used to hide the CAPTCHA code. Complete black mask is overlayed over the canvas, hence preventing OCR reading.
• JavaScript and CSS have been used to create the spotlight effect i.e. When the user hovers the mouse over certain area, that particular area below is revealed while the other part is still masked in black.
• A text input field is provided for user to input code hecould read.
• As a security measure, the canvas image is not extractable.
• On successful solving of challenge, a webpage redirection is made using jquery to the Success page

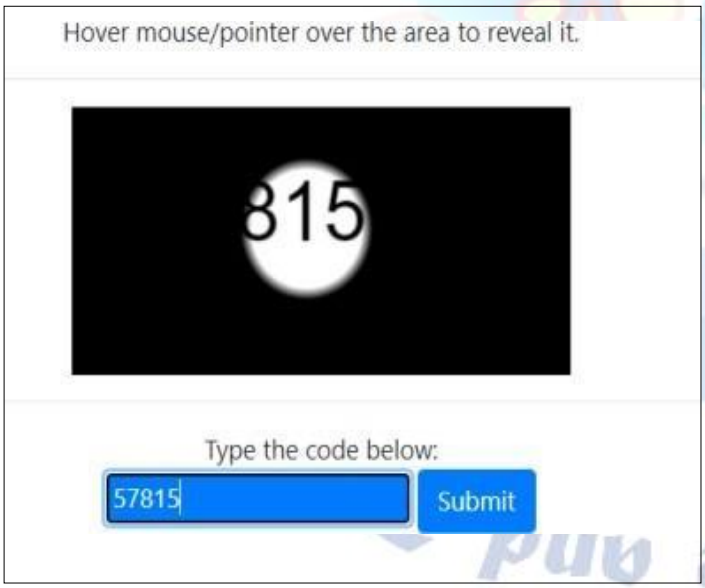


Figure 4.5: *Spotlight Captcha*

4. Sliding Puzzle CAPTCHA

Eight puzzle problem is also known by the name of N puzzle problem or sliding puzzle problem. N- puzzle that consists of N tiles (N+1 titles with an empty tile) where Ncan be 8, 15, 24 and so on.

In our example N = 8. (that is square root of (8+1) = 3 rows and 3 columns). In the same way, if we have N =15, 24 in this way, then they have Row and columns as follow (square root of (N+1) rows and square root of (N+1) columns). That is if N=15 than number of rows and columns= 4, and if N= 24 number of rows and columns= 5.

We are given an initial state or initial configuration (Startstate) and a Goal state or Goal Configuration

Initial State

1	2	3
	4	6
7	5	8

Final State

1	2	3
4	5	6
7	8	

Instead of moving the tiles in the empty space we can visualize moving the empty space in place of the tile. The empty space can only move in four directions (Movement of empty space): Up, Down, Right or Left. The empty space cannot move diagonally and can take only one step at a time

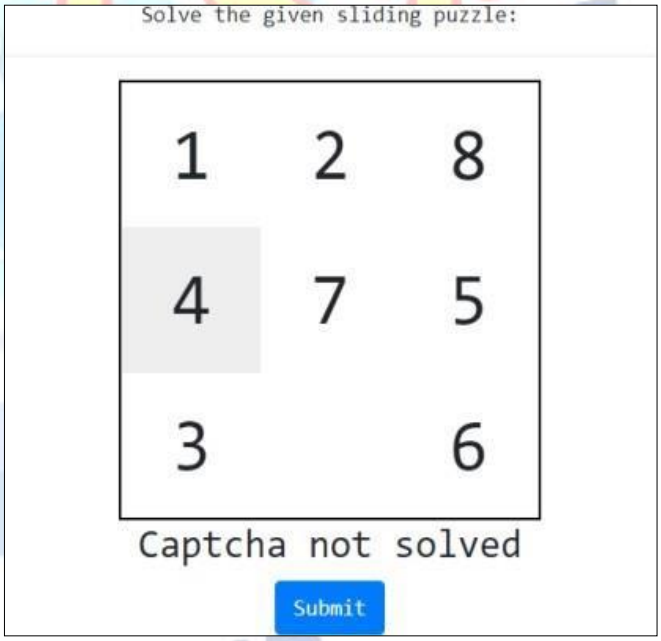


Figure 4.7: *Sliding Captcha*

5. CONCLUSION

The proposed captcha verification method provides overall better user experience. It can be used by visually impaired users as well. Color blind users as well as short-sighted users can find it easy to use.

This captcha is hard to crack as developing a bot that can draw something on canvas is almost impossible. Not

only that, questions given to the users wouldbe completely random, hence bot has to understand the question first. Hence cracking this captcha can be extremely challenging for a bot but for the users, it would be a piece of cake.

Interactive Challenge based CAPTCHA like Sliding puzzle, Spotlight puzzle and checkerboard puzzle provide following benefits over generic text-based captchas:

- Better user experience (UX) by eliminating the needto read distorted text. More engaging to users.
- No language barrier as challenge instructions (and question, if any) can be easily translated using browser tools/screen reader and the challenge itself utilizes numbers set that is universal.
- Hard to write mass spam bots for challenges that aren't text recognition based. Exploring whole state space to solve such puzzle requires computationalas well as time resource which is not feasible for mass spam bots targeting multiple websites at once.

On the downside, sliding puzzles are definitely time taking and even hard for users to solve as a CAPTCHA challenge. There is a need to strike a right balance between security and challenge complexity when it comes to choosing CAPTCHA for some website. There are 3 basic properties that CAPTCHAs must ideally satisfy:

- easy for human users to pass.
- easy for a tester machine to generate and check.
- hard for a software robot to pass.

Conflict of interest statement

Authors declare that they do not have any conflict of interest.

REFERENCES

[1] LeCun, Y., Bottou, L., Bengio, Y. & Haffner, P. (1998). Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11):2278-2324.

[2] LeCun, Y., Cortes, C. & Burges, C. J. C. (1998) The MNIST database of handwritten digits [Data files].

[3] TensorFlow. MNIST for ML Beginners. 2017.

[4] LeCun, Y.; Cortes, C.; Burges, C.J.C. The MNIST Database of Handwritten Digits. 2012.

[5] Carnegie Mellon University, CAPTCHA: Telling Humans and Computers Apart.

[6] Cire¸san, D.C.; Meier, U.; Masci, J.; Gambardella, L.M.; Schmidhuber, J. Flexible, High Performance Convolutional Neural Networks for Image Classification.

[7] Greg Mori, Jitendra Malik , UC Berkeley Computer Vision Group, Simon Fraser University, "Breaking a Visual CAPTCHA", accessed 15 November 2020.