International Journal for Modern Trends in Science and Technology, 7(12): 144-148, 2021

ISSN: 2455-3778 online
DOI: https://doi.org/10.46501/IJMTST0712025
Available online at: http://www.ijmtst.com/vol7issue12.html

OCR Captcha

Yashwant Kumar Jha | Shallu Bashambu

Department of Information Technology, MAIT, New Delhi, India

To Cite this Article
Yashwant Kumar Jha and Shallu Bashambu. OCR Captcha. *International Journal for Modern Trends in Science and Technology* 2021, *7* pp. 144-148. https://doi.org/10.46501/IJMTST0712025

Article Info
Received: 28October 2021; Accepted: 04December 2021; Published: 07 December 2021

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 such as 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, CNN, Handwritten Digit Recognition, Web Accessibility

I.INTRODUCTION

Captcha is a type of challenge-response test used in computing to determine whether or not the user is human. 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.

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. Users will be asked some basic questions like:

- What is 3+5?
- Draw the smaller number: 3 or 9.
- What is 4-3?

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, 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.

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.

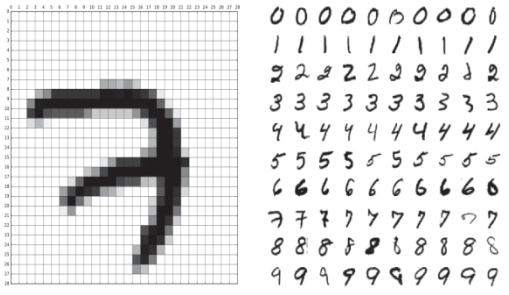
More specifically, one dataset has been widely used for this purpose: the MNIST dataset. MNIST is a database of labeled handwritten digits, with separate training and test sets, and therefore is an easily interpretable domain that allows a fast comparison between different techniques. In fact, MNIST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

METHODOLOGY

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 the 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.

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

Convolutional Neural Networks

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

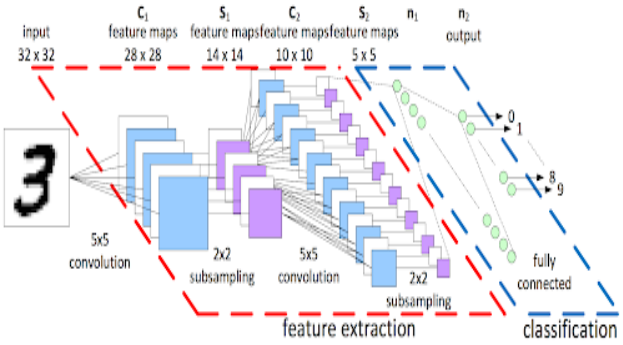


Figure 2: Convolutional Neural Network working.

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

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

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

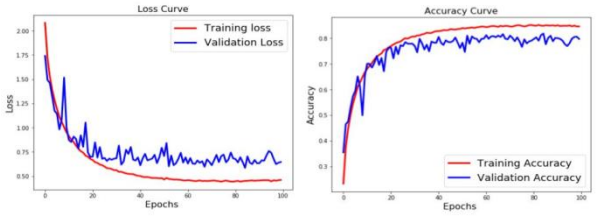
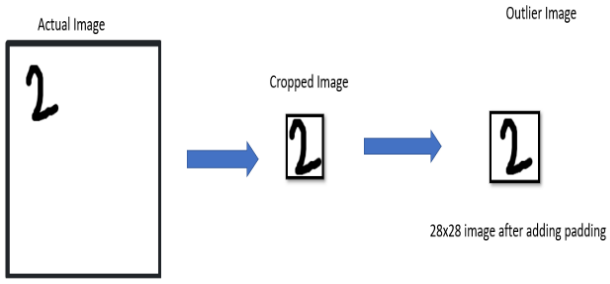


Figure 3: Convolutional Neural Network training.

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.



Proposed Workflow

Idea of is to provide the user some tasks that are 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?
- What is 4-3?

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

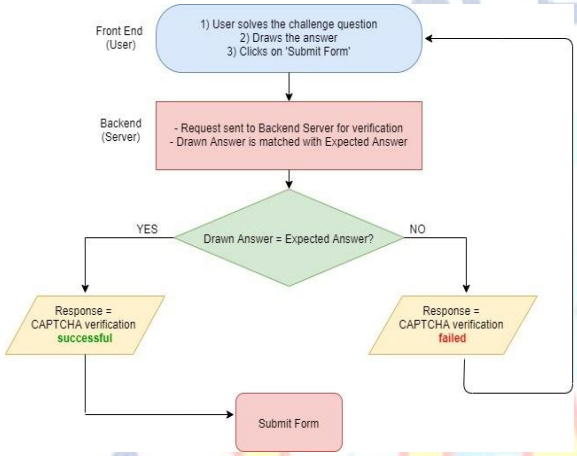


Figure 4: Workflow.

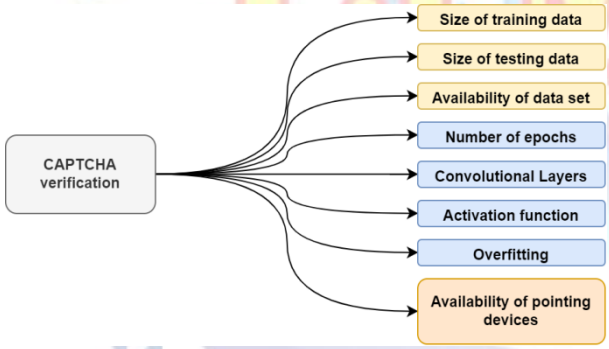


Figure 5: Dependencies.

RESULT

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 6: Training of Convolutional Neural Network.

Web Application Result

User will be asked simple mathematical questions. These questions will be generated randomly, and user will answer them by drawing the digit 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.

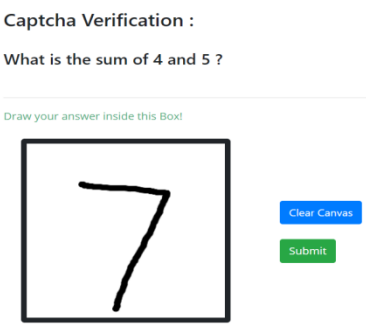


Figure 7: Web Application

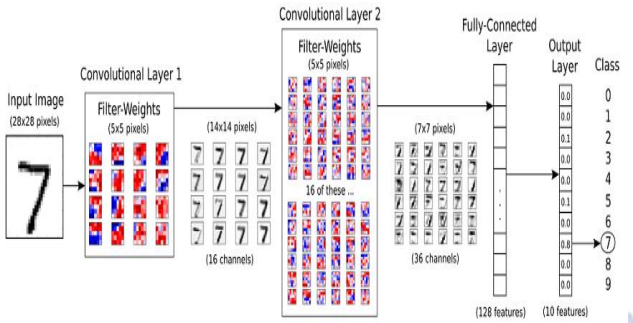


Figure 8: Working of CNN.

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 would be 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

FUTURE SCOPE AND IMPROVEMENTS

This captcha verification method is fast enough so that it can be used by all the websites for captcha verification process. Also, it provides overall better user experience.

It can be further improved by implementing following:

- Re-drawing answer and checking similarities in both drawn answers. Very close match would be rejected outright. Humans tend to have variations in drawing.
- Increasing the dataset by including alphabets as well.
- Including such questions that has multiple digit answer.

Including honeypot and other network security mechanism as secondary layers

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.