Pune Institute of Computer Technology
Dhankawadi, Pune

A SEMINAR REPORT
ON

CAPTCHA SOLVER FOR VULNERABILITY ASSESSMENT OF GOVERNMENT WEBSITE USING DEEP LEARNING

SUBMITTED BY

Name : Soham Rakhunde
Roll No. : 31168
Class: TE-1

Under the guidance of
Prof. P. S. Joshi

DEPARTMENT OF COMPUTER ENGINEERING
Academic Year 2021-22

DEPARTMENT OF COMPUTER ENGINEERING

Pune Institute of Computer Technology

Dhankawadi, Pune-43

CERTIFICATE

This is to certify that the Seminar report entitled

“CAPTCHA SOLVER FOR VULNERABILITY ASSESSMENT OF GOVERNMENT WEBSITE USING DEEP LEARNING”

Submitted by

Soham Rakhunde Roll No. : 31168

has satisfactorily completed a seminar report under the guidance of Prof. P. S. Joshi towards the partial fulfillment of third year Computer Engineering Semester I, Academic Year 2021-22 of Savitribai Phule Pune University.

Prof. P. S. Joshi
Internal Guide

Dr. M. S. Takalikar
Head of Department
Computer Engineering

Place:Pune
Date: 15/11/2021

ACKNOWLEDGEMENT

It is my pleasure to present report on ”CAPTCHA solver for vulnerability assessment of Government website using deep learning”. First of all, I would like to thank our Seminar Coordinator Prof. D. Kadam, Head of Department Dr. M. S. Takalikar and Principal Dr. R. Sreemathy for their encouragement and support.

I would also genuinely express my gratitude to my guide Prof. P. S. Joshi for her constant guidance and help. She has constantly supported me and has played crucial role in completion of this report. Her motivation and encouragement from beginning till end to make this seminar a success.

Last but not the least I would thank all the faculty,my parents and friends who have helped me.

Contents

List of Tables

List of Figures

Abstract

Completely Automated Public Turing test to tell Computers and Humans Apart abbreviated as CAPTCHA is a test to prevent bots or any other automated program from using the website/service which can be easily exploit the service with multiple requests in seconds overloading the server resulting in higher maintenance costs and making it impossible for the human users to access the service. This research aims to create an automated deep learning-based solution to crack visual CAPTCHA tests on government website vahan.nic.in which uses a rather outdated version. The aim is to investigate the weaknesses and vulnerabilities of the outdated CAPTCHA used by government institutions.

The proposed CAPTCHA solver deep learning model is a combination of Convectional Neural Network (CNN) and Recurrent Neural Network with a Connectionist Temporal Classification (CTC) batch cost function as a loss function. To make this Deep Learning even more accurate than the Generalized CAPTCHA Solver (referred from previous papers) in the least amount of training, this research specializes on solving a single type of CAPTCHA that is used in this website. With a dataset of just 10,000 labelled CAPTCHAs with over 10% manually labelled, the CAPTCHA solving accuracy of our model results leads to **99.27%** for the alpha-numerical test data-sets for Vahan website's CAPTCHA, respectively. This proves that we get higher efficiency with least amount of training data with focused deep learning algorithms. This Research concludes that a highly accurate Machine Learning Model can be created to solve obsolete CAPTCHAs. Thus Indian Government Websites every other website should switch to a newer CAPTCHA technique like reCAPTCHA for better resilience from bots, automated programs and DDOS attacks.

Keywords

CAPTCHA - Completely Automated Public Turing test to tell Computers and Humans Apart
CNN - Convolutional Neural Network
RNN - Recurrent Neural Network
CTC - Connectionist Temporal Classification
DDOS - Distributed Denial of Service
OCR - Optical Character Recognition
OpenCV - Open Source Computer Vision Library
SGD - Stochastic Gradient Descent
LSTM - Long Short Term Memory
ReLU - Rectified Linear activation Unit

1 INTRODUCTION

CAPTCHA is a tool used to differentiate between humans & computer programs. Due to this, CAPTCHA is also used to prevent cyber threats, cyber attacks, misuse, exploitation of services from the websites.

Especially, attacks are often crafted substitute humans with automated programs or bots, which try to overuse/exploit the services meant for humans. Multiple requests by the bots can cause overload on the servers and can be used to influence online polls, spamming in social media websites i.e. illegal promotion of products via spamming and also scams where thousands of spam messages are sent in hopes of targeting at least one unsuspecting user. DDoS (Distributed Denial-of-Service) is a common cyber attacks and nowadays also the Amplified DDoS attack that targets a service by overloading it with unexpected traffic to find credentials of the target or to freeze the server temporarily.

The concept of CAPTCHA is that its a method to distinguish humans & computer programs by making the user solve problem that only humans can quickly and easily solve, but the automated programs should ideally find them impossible to solve, both due to overtly high requirements of computation power and the complexity. CAPTCHAs are commonly in the form of images with numerical or alpha-numerical strings with noisy/warped text which is difficult to solve with computer vision. Theses CAPTCHAs are also in audio format for visually impaired individuals.

CAPTCHAs with various types of Protection mechanisms are shown in Fig.1. In CAPTCHA a random string of alphabets/digits or their combination are distorted using different types of protection mechanisms and noise is added over the CAPTCHA image. There are many techniques to add noise and distortion to the image to make them complex for computers yet recognisable to humans.

However, in this paper, we will focus on particular type of CAPTCHA from a government website. Due to a weak protection mechanism of this website's CAPTCHA which are slightly overlapping letters with a random yellow strikeout is pretty easy to be solved with modern day computational power using Machine Learning. The

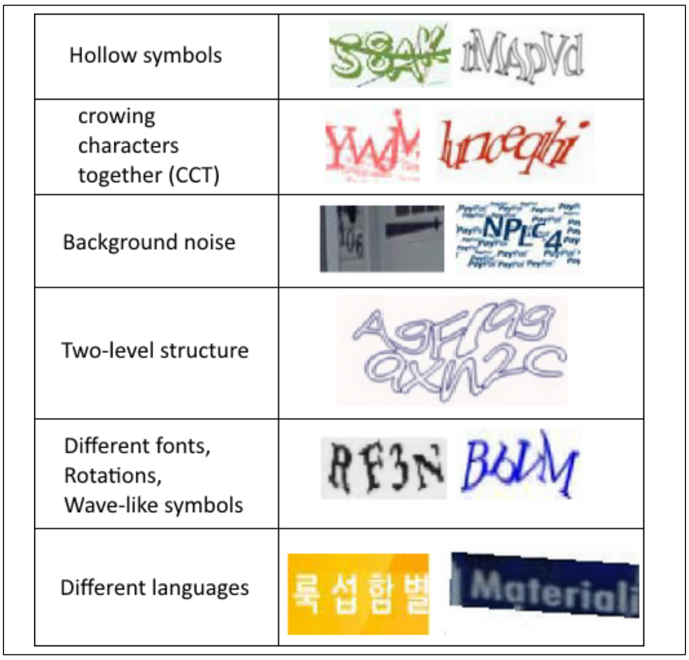

Hollow symbols
crowing characters together (CCT)
Background noise
Two-level structure
Different fonts, Rotations, Wave-like symbols
Different languages

Figure 1: CAPTCHAs with different protection mechanisms

CAPTCHAs used for this paper are showcased in Fig.2

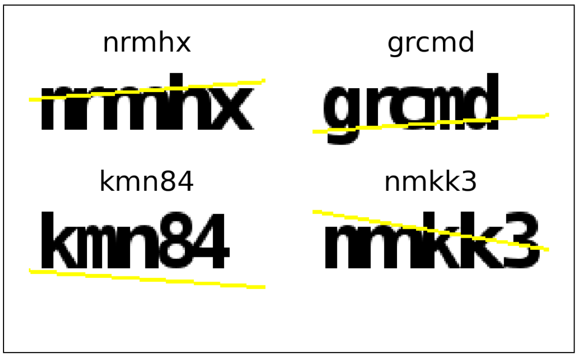


Figure 2: CAPTCHAs from Vahan site

This research tries to assess vulnerabilities of government website vahan.nic.in by developing a Machine Learning model to solve the particular type of CAPTCHA from the website with least amount of labelled training data and accuracy higher than 99% or higher.

2 MOTIVATION

With exponential growth in computational power past cyber security practices have rendered themselves obsolete. With more and more powerful technologies like Artificial Intelligence, Machine Learning, Optical Character Recognition etc. The whole internet security has evolved to prevent attacks made with the assistance of these technologies.

Thus reCAPTCHA replaced text based CAPTHCAs to protect websites from automated bots and are much resilient from Machine Learning due to wide variety of type of image based questions. But many of Indian government websites still use obsolete CAPTCHAs which can be easily solved with Machine Learning model and thus are vulnerable to DDOS attacks.

Recently in one of my previous project needed automatically scrape some data from Vahan website. But this data was protected with a login with CAPTCHA protection. But due to this CAPTCHAs design was simple and primitive enough with a simple combination of OpenCV for preprocessing and Tesseract optical character recognition (OCR) engine we could solve the CAPTCHA with approximate accuracy of 20% - 25% approximately.

Thus creating a Machine Learning model to solve its CAPTCHAs can assess the vulnerability of this website to DDOS attacks or misuse of the service provided by the website.

3 LITERATURE SURVEY

The Following table shows the literature survey by comparing techniques propose in various references:

Table 1: Literature survey

No.	Techniques	Feature considered	dataset	limitations
1	Generalised Covolutional Neural Network	Generalised for all types of CAPTCHAs with 98.31% accuracy. Uses CNN, Softmax for activation and Adam/SGD as a optimizer	Generated from Python ImageCaptcha Library	Works for 5 lettered alpha-numeric CAPTCHAs. Took 5,00,00 CAPTCHA images for training
2	CNN based Multi-labelled CAPTCHA solver	Uses a single type of generated CAPTCHA and Baidu CAPTCHA using a multi-label Convolutional neural network.	Generated from “Captcha 0.2.1” and Baidu CAPTCHA	Works with only 4 digit numeric Captcha with just 94.26% accuracy or with 4 character alphabet CAPTCHA’s accuracy is just 43.05%. Uses over 50,000 CAPTCHAs for training but still has poor results due to shortcomings of CNNs

4 A SURVEY ON PAPERS

4.1 Deep-CAPTCHA: a deep learning based CAPTCHA solver for vulnerability assessment - Zahra, Noury, Mahdi Rezaei

Authors have developed a Convolutional Neural Network (CNN) called Deep-CAPTCHA as a Generalised Captcha Solver. Their network's accuracy are **98.94%** for the numerical CAPTCHA & **98.31%** for the alpha-numerical CAPTCHA.

The paper achieved highest accuracy using Softmax activation function which was over 10% more accurate over Sigmoid activation function. They trained this neural network model on 5,00,000 randomly generated CAPTCHAs. This model was only trained for 5 character numeric and alphanumeric CAPTCHAs

4.2 A Multi-label Neural Network Approach to Solving Connected CAPTCHAs - Ke Qing, Rong Zhang

This paper focuses on training on generated CAPTCHA and Baidu's CAPTCHA and testing on the same. This paper takes two approaches the first is image segmentation and feeding segmented images to single-labeled neural networks and second is Multi label neural network which processes the whole image at once. The authors trained both of these model with 50,000 CAPTCHA images and found out the multi label to work better with accuracy of **94.26%** on "Captcha 0.2.1" which is a generated 4 digit numeric CAPTCHA and **43.05%** on Baidu CAPTCHA which is only a 4 character alphabetical CAPTCHAs but faces issue of overfitting the data.

4.3 Neural Network CAPTCHA Crackers - Geetika Garg, Chris Pollett

This paper experiments several methods of deep neural networks to break character-based for any type of image CAPTCHAs (generalised). This paper uses a combination of Convolutional Neural Network and a LSTM (Long Short Term Memory) i.e. a type of Recurrent Neural Networks instead of traditional approach to segment the image. The paper achieved accuracy of over 99.8% for fixed size CAPTHCAs and 81% on variable length CAPTCHAs.

4.4 CAPTCHA: Using Hard AI Problems for Security - Luis von Ahn, Manuel Blum, Nicholas J. Hopper, and John Langford

Authors propose Artificial Intelligence problems to create CAPTCHAs as a result of symbiosis of cryptography and Artificial Intelligence. The research introduces two families of CAPTCHAs: MATCHA and PIX. These methods provide an interesting approach to this field and concludes that either the problems remain unsolved and there is a way to differentiate between bots and humans thus serving as a CAPTCHA test or if the problems are solved then there is a way to communicate covertly on some channels and thus a beneficial research for both Cryptography and AI fields.

4.5 reCAPTCHA: Human-Based Character Recognition via Web Security Measures - Luis von Ahn, Benjamin Maurer, Colin McMillen, David Abraham, Manuel Blum

This paper is the inception of the concept of reCAPTCHA. The authors ought to harness human processing power instead of letting the user to solve a hard text based CAPTCHA they could use this human computation power to solve problems that existing Computer Vision algorithms cannot yet solve. With further versions of this, they also introduced reCAPTCHAs that didn't even require problem solving to differentiate between humans and bots just a button click can automatically validate the human users. reCAPTHCA has proved to be much better and secure to traditional CAPTCHAs.

4.6 Recognition of CAPTCHA Characters by Supervised Machine Learning Algorithms - Ondrej Bostik, Jan Kleck

The Research evaluates the accuracy of different types of CAPTCHA solving algorithms constructed with Supervised Machine Learning. This algorithms include k-Nearest Neighbors, Pattern Recognition Neural network, Feed forward neural network, Decision trees and Support Vector Machines. It concludes that all of the algorithms used here can solve CAPTCHAs with accuracy rate of around 99%. The main differentiating factor in the algorithms is the computational costs. With best algorithm amongst tested algorithms is Pattern recognition neural network as it has both good precision & low computational cost.

5 PROBLEM DEFINITION AND SCOPE

5.1 Problem Definition

To propose a focused CAPTCHA solver for assessing vulnerabilities of government website using Deep Learning CAPTCHA solver.

5.2 Scope

The factors which are crucial and need to be considered for an accurate model is to use of at least 5,000 labelled dataset of which at least 10% should be manually labelled to feed hard to solve CAPTCHAs to the network. This paper only takes in account for single type of CAPTCHA which is available and scraped off of Vahan website. This is primarily to assess how easily this CAPTCHA can be solved with least amount of dataset, this research uses just 10,240 images for training purposes.

Since the base papers generalised DEEP-CAPTCHA is less accurate than 99% with training from 5,00,000 Generalised CAPTCHAs. This paper aims to improve this and achieve accuracy of over 99.3% accuracy with training dataset of just 10,240 specific-type of CAPTCHAs.

Also other paper Multi Label CAPTCHA solver also uses a specific type of CAPTCHA and only achieves 94.26% only. This paper improves this accuracy substantially i.e. 99.3% accuracy by using combination of CNNs and RNNs unlike previous paper that only used CNNs.

6 METHODOLOGY

Deep learning applications are used in almost all aspects of our life, like autonomous vehicles, surveillance, Robotics, smart devices, ad recommendations algorithms. To solve the CAPTCHA we have also used a deep neural network architecture using convolutional layers and recurrent neural network. Below, we describe the detailed architecture of the Neural Network for solving the CAPTCHAs. The process includes pre-processing input image, decoding the output, calculating loss and accuracy because of the multi character nature of output which as a whole needs to be correct and accuracy depends on all the characters of CAPTCHA being correctly recognised.

6.1 Data Preparation

The CAPTCHA images have been custom scraped and labeled for this paper. The Dataset is created and converted into batches of data. Each batch consist of 16 Tensors comprising of CAPTCHA image, label, input shape, and label shape. Batching reduces the time to train the Neural Network. This is due to parallel computation of whole batch is much more efficient due to GPUs (Graphic Processing Units).

The images/dataset used for this research were custom scraped off from Vahan website and 85 to 90% were auto labeled using OpenCV and Tesseract OCR. The remaining 10 to 15% were manually labeled to include some of the hard to solve CAPTCHAs. 10,240 images are used for training the network, of which 1,300 were manually labeled. Training dataset of less than 5,000 could cause the problem of overfitting. 1,104 images were use for testing/validation of the Neural Network of which 160 were manually labeled.

6.2 Preprocessing

We start by applying pre-processing operations like colour space conversion (RGB to Grey scale), and noise reduction by using OpenCV to filter the image which can drastically increase the overall accuracy of the final output. The original shape of the CAPTCHA image used for training is 126 × 45 × 3 pixel i.e. three color channels (RGB). The color data in CAPTCHA is not usually useful and can inversely affect performance and accuracy of the network.

Due to this we change this image from RGB color space to a Grey-scale color space which only stores one color channel. The Greyscale image also removes the yellow protection line replacing it with blank space that obscures part of the CAPTCHA image as shown in Figure 2. Thus the final size after preprocessing are 126 × 45 × 1 pixel. This size reduction tremendously decreases the training time since it reduces the data to be processed without much loss of usable information.

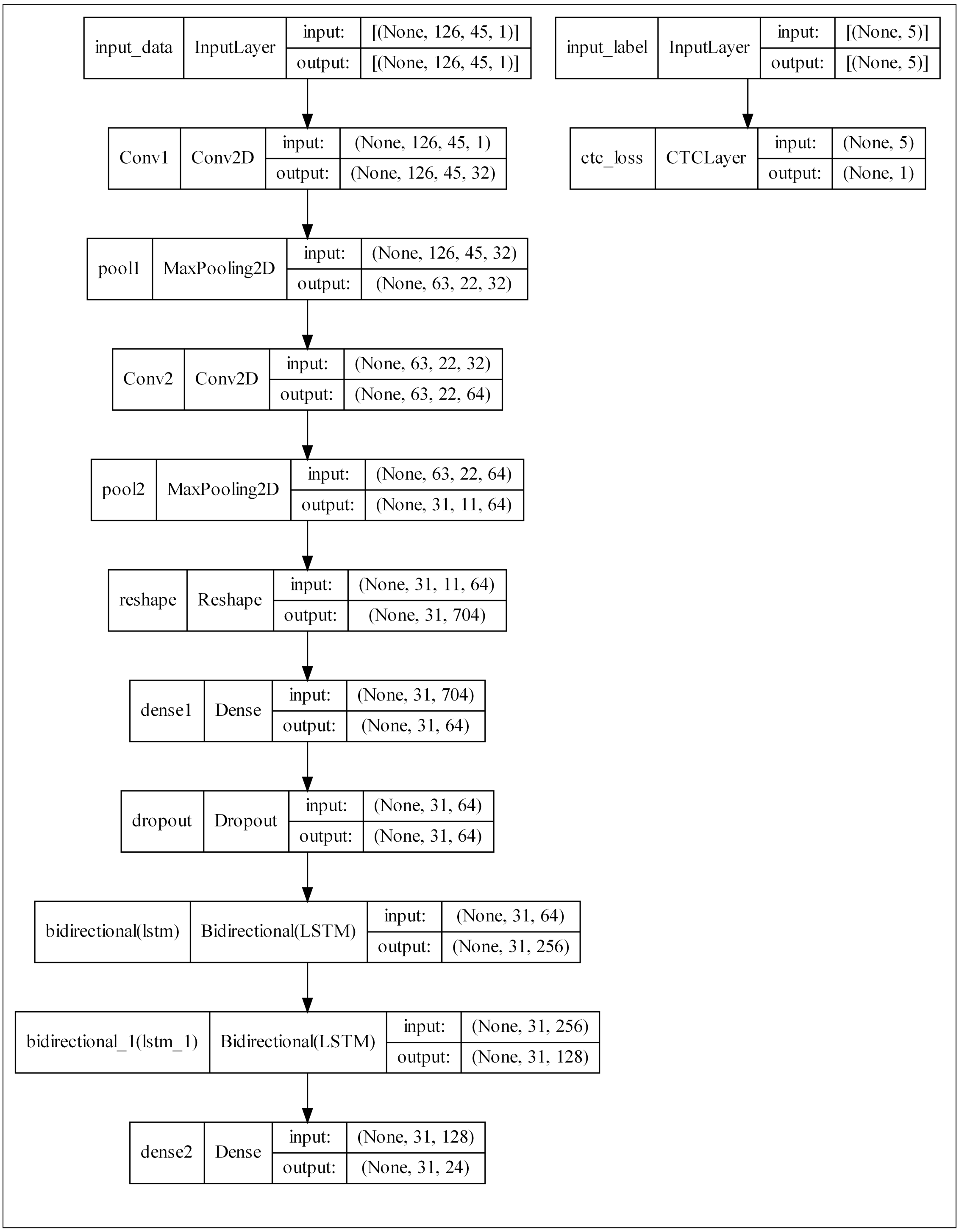

input_data | InputLayer | input: | [(None, 126, 45, 1)] | output: | [(None, 126, 45, 1)]
input_label | InputLayer | input: | [(None, 5)] | output: | [(None, 5)]
Conv1 | Conv2D | input: | (None, 126, 45, 1) | output: | (None, 126, 45, 32)
ctc_loss | CTCLayer | input: | (None, 5) | output: | (None, 1)
pool1 | MaxPooling2D | input: | (None, 126, 45, 32) | output: | (None, 63, 22, 32)
Conv2 | Conv2D | input: | (None, 63, 22, 32) | output: | (None, 63, 22, 64)
pool2 | MaxPooling2D | input: | (None, 63, 22, 64) | output: | (None, 31, 11, 64)
reshape | Reshape | input: | (None, 31, 11, 64) | output: | (None, 31, 704)
dense1 | Dense | input: | (None, 31, 704) | output: | (None, 31, 64)
dropout | Dropout | input: | (None, 31, 64) | output: | (None, 31, 64)
bidirectional(lstm) | Bidirectional(LSTM) | input: | (None, 31, 64) | output: | (None, 31, 256)
bidirectional_1(lstm_1) | Bidirectional(LSTM) | input: | (None, 31, 256) | output: | (None, 31, 128)
dense2 | Dense | input: | (None, 31, 128) | output: | (None, 31, 24)

Figure 3: Architecture of neural network model

6.3 Encoding

By analyzing all the labels the dataset contains of 22 unique alphanumeric characters. Unlike the classification problems where the deep learning model have to predict a single result from a specific number of classes, Meanwhile in the CAPTCHA solver the output is combination of multiple such classification problems i.e. one for each character in CAPTCHA. Hence, for a CAPTCHA solver model with five alphanumeric characters with 22 possible characters for each character, we get around 5 million unique combinations. This is why, we are required to encode the input labels, assigning each character a number and vice-versa for decoding.

Unlike classification algorithm the loss and accuracy of the model is dependent on all the characters of CAPTCHA combined. To achieve this we use CTC (Connectionist Temporal Classification) batch cost to calculate the loss for the neural network. A custom callback is also created to calculate the accuracy for each Epoch while training. Since individual accuracy for each digit of CAPTCHA is worthless.

6.4 Architecture of Network

Although using just CNN can result in high accuracy. Using it in combination with RNNs helps interpreting temporal information such as sequence of individual characters in CAPTCHA.

The architecture of neural network model is shown in Figure 3. This network starts with Convolutional layers with 32 neurons, & a ReLU activation function and 3 × 3 Kernel size. Followed by a 2 × 2 Max-Pooling layer. Then, we again have a similar Convolutional - MaxPooling pair with same parameters except for the number of neurons, which is 64 neurons. By the end of these layers 126 × 45 × 1 input turns into a much smaller 31 × 11 × 64 tensor.

After Convolution layers it is followed with a reshape layer to convert its input from 3D to 2D numpy array. Followed by a 64 dense layer with the ReLU activation function, and then a Dropout layer with 20% drop rate, this layer drops some of the nodes from neural network.

Now the network is further connected with a 128 connected Bidirectional LSTM RNN layer with a dropout of 20% and tanh activation. Then we again have a similar 64 connected Bidirectional LSTM RNN layer with a dropout of 25%.

After LSTM layers this is connected to 23 Dense layer with a softmax activation. 23 signifies one for each possible character in the CAPTCHA.

Now to calculate the loss the previous Dense layer, input labels, input length and label length is passed as an Input for a Custom layer that calculated the loss using CTC batch cost. Connectionist Temporal Classification(CTC) is an algorithm used to calculate loss for sequence problems like CAPTCHA solving and Handwriting recognition. And to calculate accuracy at each epoch end a custom callback class initiated a prediction for each batch to calculate accuracy as positive only if all characters in CAPTCHA.

Finally we used Stochastic Gradient Descent(SGD) optimiser due to its capability of training the model in reasonable and comparable time to Adam Optimiser.

As can be inferred from Figure 4 and Figure 5 that training for 10 epochs with a batch size of 16 for each, and epoch 5 to 10 is where both loss and accuracy starts to plateau which proves that the CAPTCHA used in this website are not very well designed.

7 EXPERIMENTATION RESULTS

We trained the proposed neural network on 10,240 scraped CAPTCHAs, of which 1,300 were manually labeled. 1,104 images were use for testing/validation of the Neural Network of which 160 were manually labeled. Note: Precision is the metric that is calculated with matching every character in the CAPTCHA. While Accuracy is calculated with matching correctly predicted CAPTCHA.

Table 2: Accuracy, Precision and Loss Metrics for the Training dataset & Testing dataset respectively.

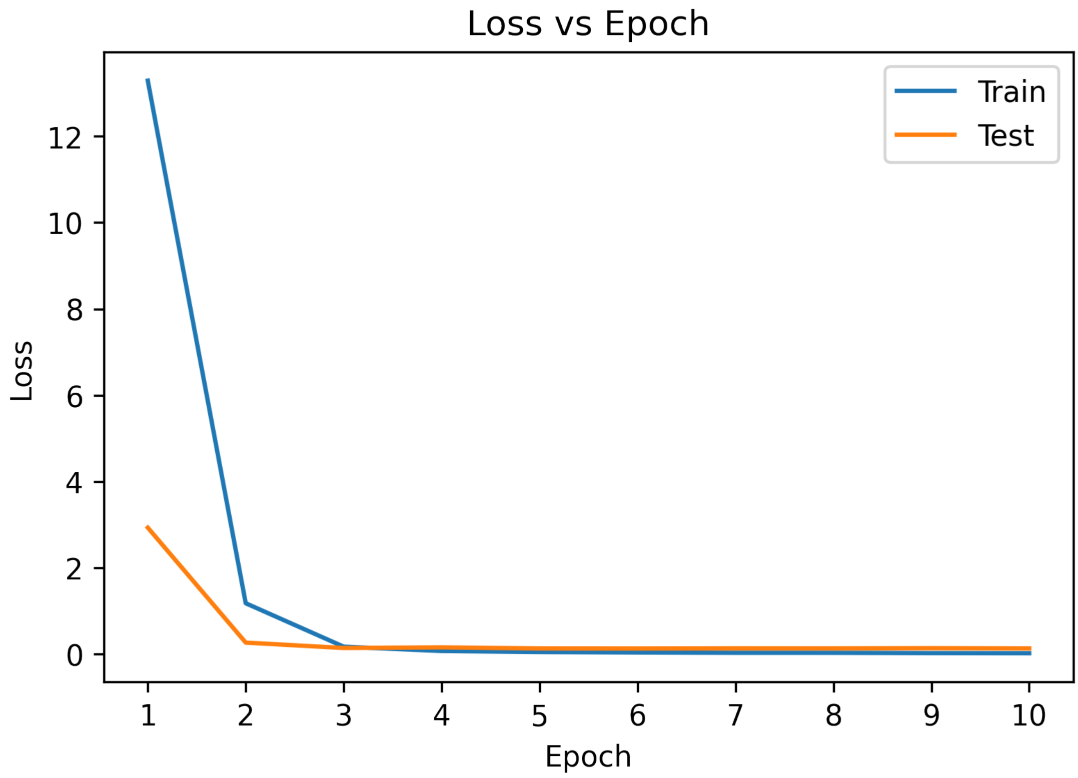
	Loss	Accuracy	Precision
Train	0.0215	99.85%	99.95%
Test	0.1324	99.27%	99.85%

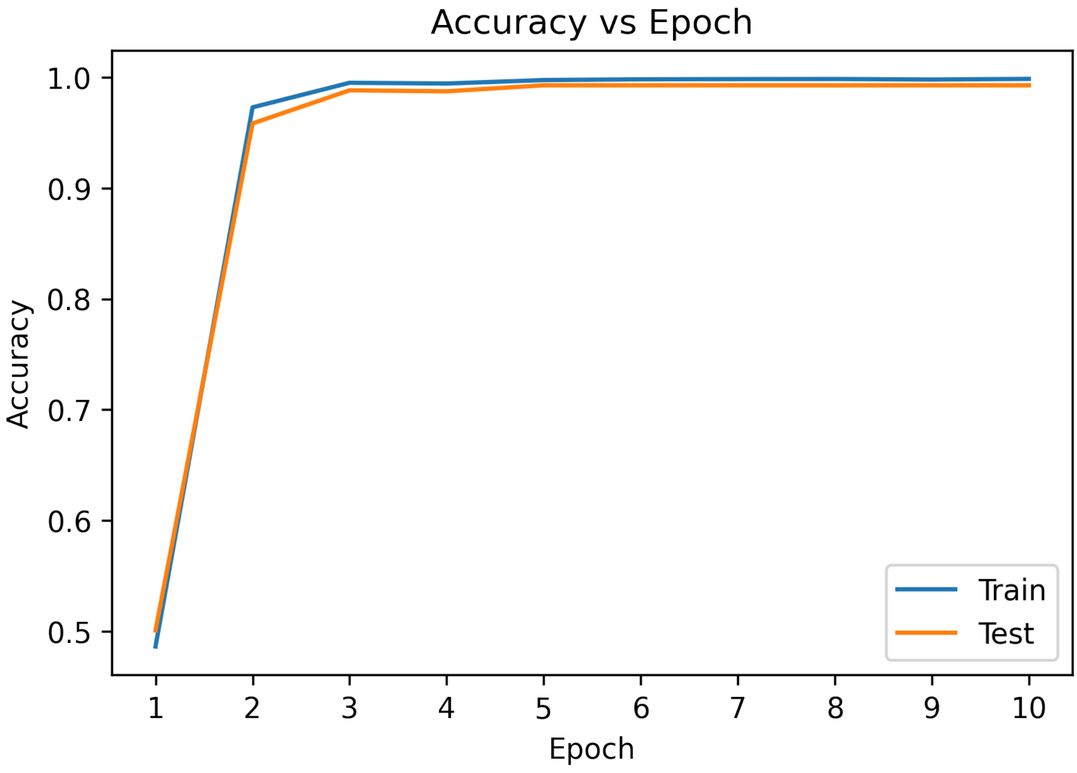
Table 3: Accuracy metric for the Training dataset & Testing dataset respectively for each digit and the accuracy of the whole CAPTCHA.

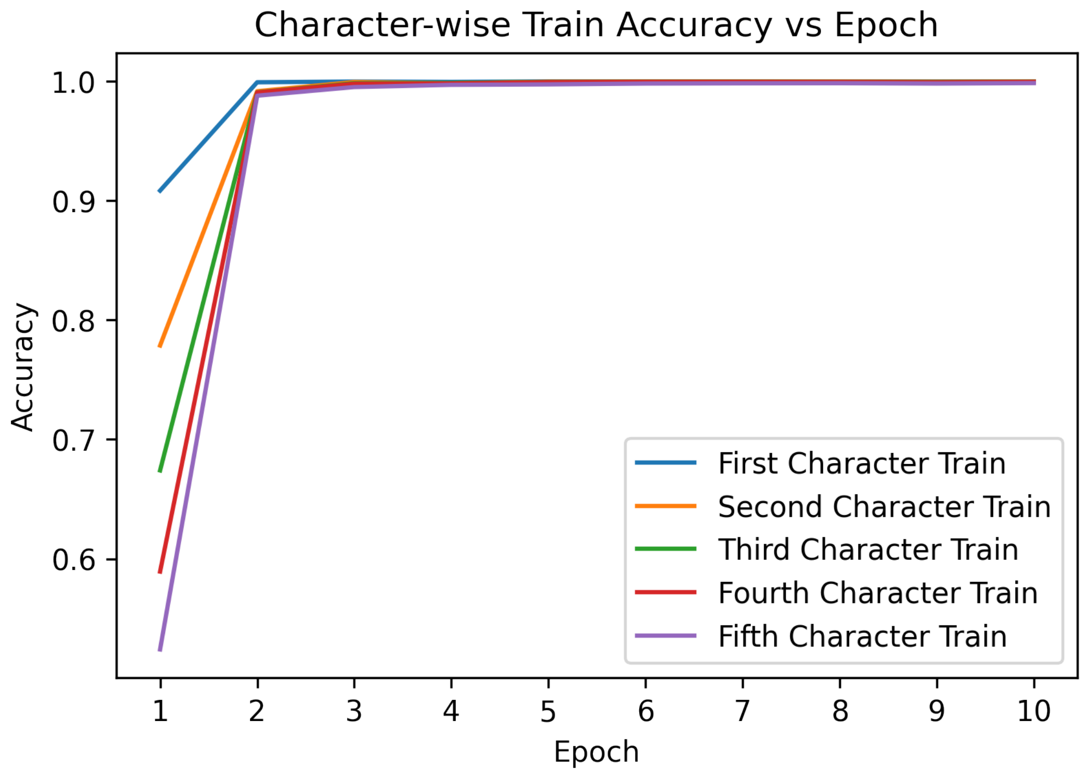
	Train Accuracy	Test Accuracy
I Character	100.0%	99.81%
II Character	99.99%	99.81%
III Character	99.97%	100.0%
IV Character	99.96%	99.72%
V Character	99.86%	99.90%
CAPTCHA	99.85%	99.27%

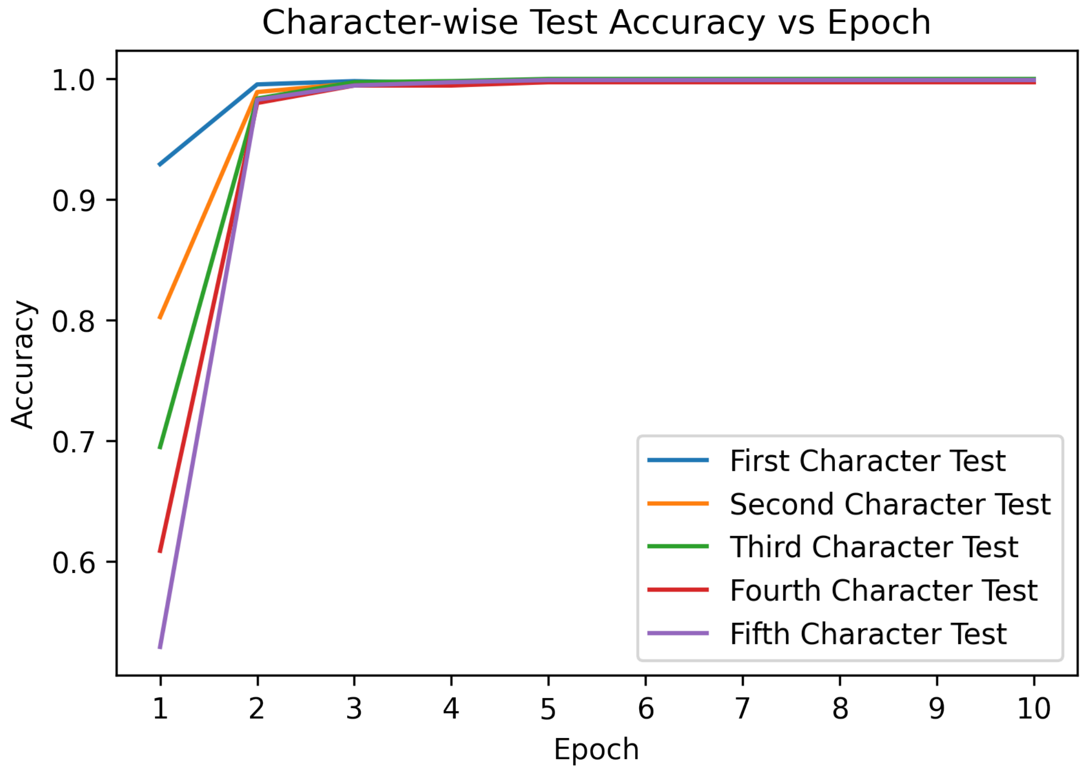
7.1 Performance Analysis

As show in Table 2, neural network achieved accuracy rate of 99.85% on the training set and 99.27% on the test dataset. Note that the accuracy metric is calculated on the basis of correctly detected CAPTCHAs as a whole (i.e. correct detection of all five individual characters in a CAPTCHA); otherwise, the accuracy pf individual characters is much greater than overall accuracy as inferred from Table 3


Loss vs Epoch
Loss
0
2
4
6
8
10
12
Epoch
1
2
3
4
5
6
7
8
9
10
Train
Test


Accuracy vs Epoch
Accuracy
0.5
0.6
0.7
0.8
0.9
1.0
Epoch
1
2
3
4
5
6
7
8
9
10
Train
Test


Character-wise Train Accuracy vs Epoch
Accuracy
1.0
0.9
0.8
0.7
0.6
Epoch
1
2
3
4
5
6
7
8
9
10
First Character Train
Second Character Train
Third Character Train
Fourth Character Train
Fifth Character Train


Character-wise Test Accuracy vs Epoch
Accuracy
1.0
0.9
0.8
0.7
0.6
Epoch
1
2
3
4
5
6
7
8
9
10
First Character Test
Second Character Test
Third Character Test
Fourth Character Test
Fifth Character Test

By finding the incorrect predictions and inspecting them we found out some important conclusion.

While a human can solve almost all of the incorrectly predicted CAPTCHAs, we found following strengths & limitations were identified in the proposed model that resulted in confusing the model to misclassify and produce incorrect results. These are as follows:

Strengths:

- The model was successfully able to overcome the protection feature, i.e. Random yellow strike-through line which obscures the CAPTCHA as shown in Figure 2.
- It is also successful with the overlapping characters for almost all characters except for rn recognised as m.
- This model is much more accurate than just using a CNN without an RNN layer as shown in Figure 4.
- It just needed 10,000 training dataset to achieve this accuracy meanwhile previous papers with generalised CAPTCHA solvers required to train on dataset of 5,00,000 images.

Weakness:

- In 78% of the misclassified samples, the model failed to output the last character because these characters were partially out of the box which made it impossible to retrieve any meaningful data for the last character.
- In the remaining 22%, the characters were overlapping causing confusion, especially the overlapping characters rn is misclassified as m as shown in Figure 4.
- This model is trained for only one type as shown in Figure 2; of 5 character alpha numeric CAPTCHA.

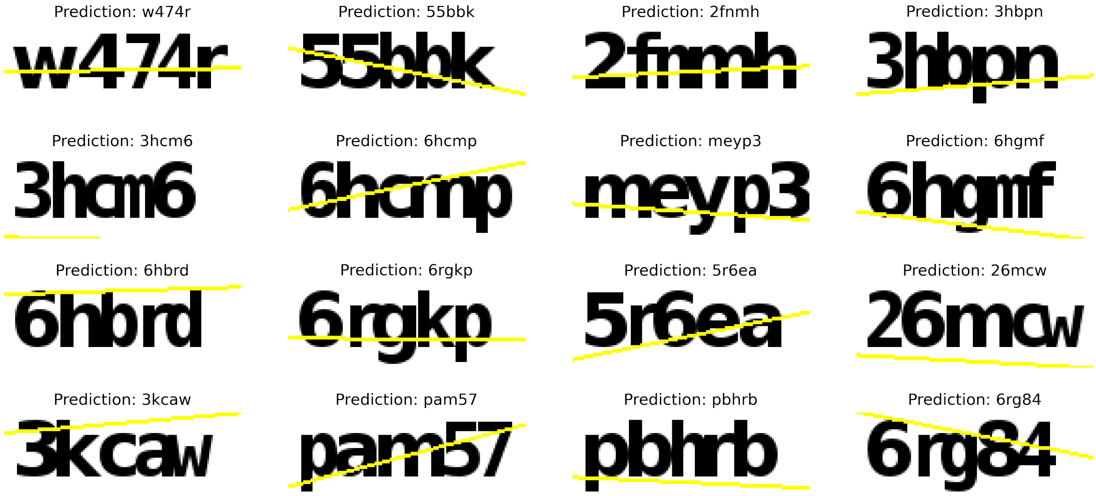


Figure 8: Correctly predicted challenging CAPTCHAs with overlapping characters

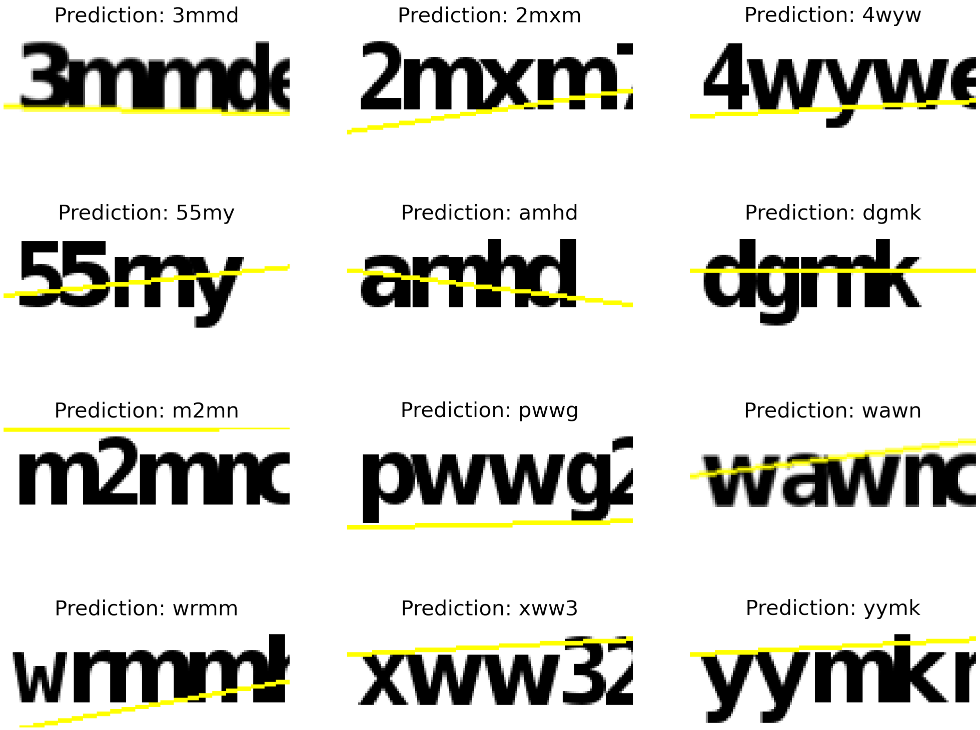


Figure 9: Incorrectly predicted CAPTCHAs

8 CONCLUSION

We developed and tuned CNN and RNN based deep neural network for 5 character alphanumerical based CAPTCHA detection from the vahan website to reveal the strengths and weaknesses of the government website. We achieved up to 99.27% accuracy as described in Table 2. Also we can infer from Figure 4 and Figure 5 that training for 10 epochs with a batch size of 16 for each, and epoch 5 to 10 is where both loss and accuracy starts to plateau which proves that the CAPTCHA used in this website are not very well designed. And since we could eaily achieve accuracy of 99.27% with just 10,000 training images it supports our argument that government websites CAPTCHAs aren't well designed by today's standards and can be easily solved with a good Deep Learning Model with minimum amount of CAPTCHA data and the newer technologies have rendered this type of CAPTCHAs obsolete.

This concludes the Indian Government Websites are highly vulnerable to DDOS attacks and misuse by automated programs. Thus, every website should switch to a newer CAPTCHA technique like Google's reCAPTCHA for better resilience from bots, automated programs and DDOS attacks.

References

[1] Noury, Zahra & Rezaei, Mahdi. (2020). Deep-CAPTCHA: a deep learning based CAPTCHA solver for vulnerability assessment. 10.31219/osf.io/km35b.

[2] K. Qing and R. Zhang, ”A Multi-Label Neural Network Approach to Solving Connected CAPTCHAs,” 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR), 2017, pp. 1313-1317, doi: 10.1109/ICDAR.2017.216.

[3] G. Garg and C. Pollett, ”Neural network CAPTCHA crackers,” 2016 Future Technologies Conference (FTC), 2016, pp. 853-861, doi: 10.1109/FTC.2016.7821703..

[4] von Ahn L., Blum M., Hopper N.J., Langford J. (2003) CAPTCHA: Using Hard AI Problems for Security. In: Biham E. (eds) Advances in Cryptology — EUROCRYPT 2003. EUROCRYPT 2003. Lecture Notes in Computer Science, vol 2656. Springer, Berlin, Heidelberg. https://doi.org/10.1007/3-540-39200-9_18.

[5] Ahn, Luis & Maurer, Benjamin & McMillen, Colin & Abraham, David & Blum, Manuel. (2008). reCAPTCHA: Human-based character recognition via Web security measures. Science (New York, N.Y.). 321. 1465-8. 10.1126/science.1160379.

[6] Boštík, Ondřej & Klečka, Jan. (2018). Recognition of CAPTCHA Characters by Supervised Machine Learning Algorithms. IFAC-PapersOnLine. 51. 208-213. 10.1016/j.ifacol.2018.07.155.