**CAPTCHA Recognition using Convolutional Neural Networks**

**Manvi Goel Nandini Sidana Siddharth Samber**

**CO17336 CO17339 CO17358**

***Abstract***

CAPTCHAs may be referred to those infuriating images containing the text that needs to be typed in before a person can access a particular website. The full form of CAPTCHA is "Completely Automated Public Turing test to tell Computers and Humans Apart [3]" and as the name suggests it is a way to avert the computer to fill out the form on its own, automatically and hence verifying if the person filling the form is real and therefore is also known as Human Interactive Proofs (HIP) [2].However using the concept of deep learning and computer vision, the very purpose of the CAPTCHAs can be defeated. This test can be passed automatically with the help of Convolutional Neural networks - a class of the deep neural networks. The breaking of the CAPTCHA decides a system's security and also acts as a measure of how reliable a system is [2]; hence research in this field is a matter of great concern. In this paper, we have created a model that can break a CAPTCHA. We have used a dataset of CAPTCHA images where each image is a five-character CAPTCHA containing only lower case English letters and digits. Firstly, we have pre-processed this dataset to remove noise and hence obtain grayscale images and the target array containing information regarding the characters present in each CAPTCHA image. Following this we have trained the model using Convolutional Neural Networks and a detailed description of how the model is trained is proposed. Our model consists of 24 layers and outputs layer wise accuracy. We have trained the model for 60 epochs and it has been observed that as the number of epoch increases the loss decreases and hence the efficiency of the model increases.

***Keywords:*** CAPTCHA ***Recognition, Convolutional Neural Network, Keras, Input layer, Pooling layer, Dense layer, Flatten layer, Deep learning*.**

1. **INTRODUCTION –**

CAPTCHA stands for Completely Automated Public Turing test to tell Computers and Humans Apart. It is multimedia security mechanisms which help us to distinguish between humans and computers. In order to verify the security and reliability of CAPTCHA many CAPTCHA breaking technologies were invented that include image processing, pattern recognition, computer vision, and artificial intelligence. Main challenges which we face in text based CAPTCHA are large character set, distortion of characters ,overlapping of characters, adhesion of characters, dealing with different length, size, angle, font of characters, unfixed length string , hollow characters, color and shape of complex background.

For understanding the process of text based CAPTCHA breaking techniques we need to understand some concepts. Segmentation is one of them. In segmentation we segment individual characters or components of individual characters from text, this is known as segmentation. Now there are different techniques for breaking text based CAPTCHA. We can classify text based CAPTCHA breaking techniques in two based on whether we use segmentation or not. Success rate for breaking methods based on segmentation is high when we don’t use adherent characters in our CAPTCHA. Moreover anti-segmentation techniques have been found for which this breaking method using segmentation is of no use. To solve that we have a Breaking method based on non-segmentation, these are good when we have adherent characters in our CAPTCHA. General Framework for text based CAPTCHA is preprocessing segmentation, combination, recognition, post processing .Its main purpose is to highlight the information related to characters in a given image and to weaken or eliminate interfering information. It is majorly of three types: image binarization and image thinning, Image De-noising. Image binarization is a technique to highlight interesting objects’ contour and to remove noises in the background. The key to image binarization is to select an appropriate threshold. Image thinning is to process the character’s contour as a skeleton .It must not change the character’s adhesion. It is performed either by iterative or recursive method. In order to resist breaking, there are noises and interference lines in CAPTCHA images. In addition, some noises are generated during grayscale and binarization. Therefore, we need to denoise the CAPTCHA image. The technique to de-noise this is Image De-noising. Segmentation is breaking text into individual characters or individual character components. We have two types of segmentation, segmentation based on individual characters and segmentation based on Character Components. Segmentation based on individual characters can be achieved using segmentation Methods Based on Character Projection, Connected Components, Character Width, Character Feature etc. And segmentation based on Character Components can be achieved using segmentation Methods Based on Character Structure etc. Segmentation based on character components breaks the text into character components and not characters. Therefore we need to combine these components into a character that is called Combination. There are basically two types of combination, combination based on redundancy and non-redundancy. Recognition method is used to recognize the individual characters. The recognition methods based on machine learning is basically divided into three categories. First one is the traditional methods in which we use algorithms like SVM and KNN classifiers to recognize characters. Second is the methods based on neural networks which use neural networks, however when applying neural networks we need to extract features first. Third is the method based on deep learning, the methods commonly used here are CNN, RNN, LSTM-RNN etc. In this we recognize images without extracting features.

1. **RELATED WORK**

In [2], a detailed review of the latest developments in the text-based CAPTCHA breaking field has been given. Further in the paper, a framework of text-based CAPTCHA breaking technique has been proposed. The framework in this paper mainly consists of segmentation, combination, preprocessing, post processing and some other modules; and all these modules are discussed at length, basically talking about the technique research progress of the technique involved in each module, comparing and analyzing the different methods of segmentation and recognition. Last but not least, a few problems relating to CAPTCHA have also been discussed.

In [3], an active deep learning strategy that makes use of the ability to gain new training data for free without any human intervention which is possible in the special case of CAPTCHAs. The way to choose the new samples to re-train the network and present results on an auto-generated CAPTCHA dataset has also been discussed. The approach proposed in the paper improves the performance of the network to a great extent if only few labeled training data has been taken into consideration initially.

In [7], an effort to improve dense net is made .Dense net for CAPTCHA recognition is created which reduces the number of convolution blocks and then this DCFR model is used to test the English and Chinese text CAPTCHAs with different numbers of characters. It reduces memory consumption which is a disadvantage of Densenet while maintaining the excellent classification performance of dense net.

In [8], a focus is set on capturing the intelligent spywares that can break through the New CAPTCHA and hence trap IDS (Intrusion Detection systems) in order to gather information about the same and the necessary actions that can be taken against it. Also a security model has been designed in this paper that has CAPTCHA IDS with a redirector, IPS (Intrusion Prevention System) and a honeypot that can detect intrusion by intelligent spyware. This model ensures security against intrusion by spywares.

In [9], a number of recognition methods are analyzed and one of them based on Recurrent Neural Network (RNN) is brought up; this specific method is composed of long short-term memory blocks, known as LSTM blocks. The feature extraction for CAPTCHA recognition has been studied in detail and it has been concluded that gray value of an image proves to be an efficient feature for RNN, Also, an algorithm has been proposed to improve recognition rate and lower the time complexity.

In [10], a deep learning technique of Convolution neural networks (CNN) for breaking a text based CAPTCHA with faster speed and moreover with high accuracy rate is implemented proving that text based CAPTCHAs no longer provide security. Text CAPTCHAs are given several distorted letters so that humans are able to solve them easily within a small amount of time but It is hard for machines to recognize. But with the rapid enhancement of technology in artificial intelligence, hackers are trying to break the security by breaking the CAPTCHAs with faster rate and high accuracy.

In [12], Multi letter CAPTCHA recognition and single letter CAPTCHA recognition methods have been explained and CAPTCHAs are generated using PyCAPTCHA python package to generate CAPTCHAs. In this paper Ml based CAPTCHA breaker is made that maps CAPTCHA to their solution. Different techniques for CAPTCHA recognition have been proposed.

In [14], an attempt to hack the CAPTCHA is made. To be precise, the touclick Chinese CAPTCHA has been targeted which is used in mobile application scenarios. To hack such a CAPTCHA is quite challenging. A multi-scale Corner based Structure Model, known as CSM, with a very efficient pattern matching scheme, which can accurately capture the touclick Chinese CAPTCHA is used. A demonstration of the effectiveness and efficiency of the proposed approach by extensive experiments on a Chinese touclick CAPTCHA dataset has also been given. Some suggestions on improving the current CAPTCHA-based human-machine identification systems have also been given.

1. **CONVOLUTIONAL NEURAL NETWORK**

An artificial neural network is an attempt to replicate the neural structure of the human brain [6]. It can be thought of as a human brain where different hidden layers play the role of the neurons to process the input and transmit this processed input information to produce the output using a specified algorithm. It is a mesh of various interconnected nodes, and each node symbolizes an artificial neuron. These neural networks can be classified into a number of categories depending on the input type. The class of deep neural networks applied to analyses the visual imagery is the Convolutional Neural Networks, also known as ConvNets or CNNs; the model under study in our project.

A CNN is an algorithm of deep learning which takes in an image as input and then assigns some value to various features in the image which further helps to differentiate one feature from the other. When compared to other classification algorithms, the preprocessing required in CNN is very low. The main purpose of a Classification Neural Network is to transform the images into a form which is much easier to process, without losing aspects/features which are essential for getting an optimized prediction. Talking about the architecture of a CNN, the neurons in its layers are arranged in 3 dimensions viz. height, width and depth. For our project, we have used the CNN model for CAPTCHA recognition. The input image passes through a series of different types of layers before the final output is obtained. These generic layers are:

* Input Layer: This is the first layer used in a Convolutional Neural network. This layer is used to take an image as input; also to instantiate a Keras tensor, the input layer is used.
* Convolutional Layer: This layer is also known as *the kernel*. It is the foundation layer of CNN; the layer which extracts the input features from the image. These networks are not limited to one such single layer. There can be various convolutional layers with the first layer extracting the low level features from the image and the consecutive layers extracting the high level features from the image.
* Pooling Layer: It is a form of non-linear down sampling; performing dimensionality reduction, it reduces the computational power required to process the data. There are two types of pooling: Average Pooling and Max Pooling. To be specific, in our project we have specifically performed *max pooling,* which is the most common function to implement pooling. The Max Pooling layer returns the maximum value from the image part covered by the convolutional layer. This layer performs dimensionality reduction and denoising, and hence creates feature maps that summarize all the input features.
* Flatten Layer: The pooled feature map obtained from the max pooling layer is flattened in this layer, i.e. converting it into a long vector suitable enough to be easily processed further by the artificial neural network, thus making back propagation easier.
* Dense Layer: The dense layer computes the dot product between the input and the kernel along the last axis of the inputs and axis 1 of the kernel.
* Dropout Layer: This layer is mainly a regularization technique; it is added to prevent the over-fitting of the model. It is really necessary to do away with the over fitting because an over fitted model has an extended capacity to learn the noise in the observation and To accommodate the noise, such model overstretches itself and ignores domains not covered by data. As a result of which, poor predictions are made by the model outside the domain of the training set.

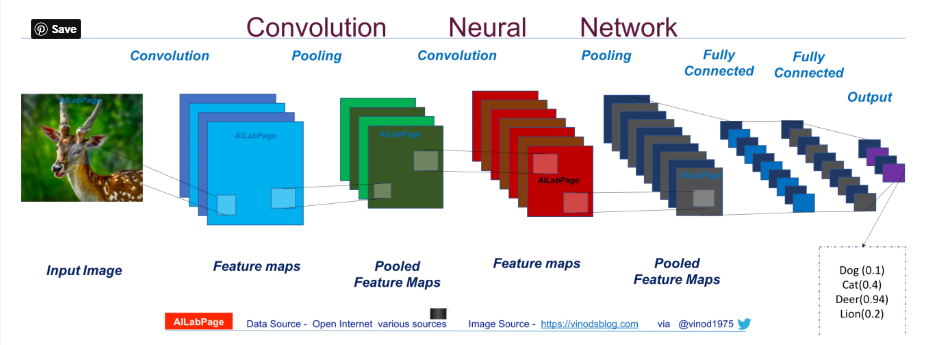


Figure 1 Convolutional Neural Network [5]

1. **TOOLSET USED:**
   1. **Python 3:**

It is a programming language with many in-built libraries for deep learning, computer vision and so many other applications that make building projects so easier and fun to do.

* 1. **OpenCV:**

It is a python library with many programming functions with the aim to provide tools which help solve the real time computer vision problems easily. The OpenCV library contains both the low-level image-processing algorithms and high-level functions such as face detection, feature matching, pedestrian detection, and tracking [1]. In our project, we have used this library to process the CAPTCHA images.

* 1. **Keras:**

It is an open source library in python used for building neural networks [4]. Keras speeds up the related processes to a great extent. It reduces the number of user actions that are needed for common use cases, and it gives clear, comprehensible and actionable error messages. This library is mainly used for either convolutional networks or recurrent networks or a combination of both [4].

* 1. **os:**

This module provides various functions required to interact with the operating system. These functions work independently of the operating system type be it Windows, Mac or Linux; all these functions work the same way on all the platforms.

1. **DATASET DESCRIPTION:**

The dataset used for this project consists of 1070 .png images of text based CAPTCHA. The dataset has been taken from [https://www.researchgate.net/publication/248380891\_CAPTCHA\_dataset](https://www.researchgate.net/publication/248380891_captcha_dataset). Each dataset image is of 5 character set and the character set is defined as all English small letters and digits from 0 to 9. Hence a total of 36 characters are present in the character set .The dimensions of the image are 50\*200 that is its height is 50 and width is 200. The CAPTCHA images consist of noise in the form of lines and blurriness. The characters in CAPTCHA images are also not straight and clear.



Figure 2 CAPTCHA Image

To train the model to read this CAPTCHA efficiently, 970 images are used for training purposes and the remaining 100 images are used for testing purposes. The noise from the dataset is removed during preprocessing when the image is converted into grayscale. Thereafter, for training the data Convolutional Neural Network is used.

1. **PROPOSED METHOD:**

The model is developed for CAPTCHA recognition of five character CAPTCHA containing only small English letters and digits. The system is trained using Convolutional Neural Network for detecting the CAPTCHA present in the image. For this purpose, the training sample dataset is first pre-processed and then the model is developed consisting of twenty four layers.

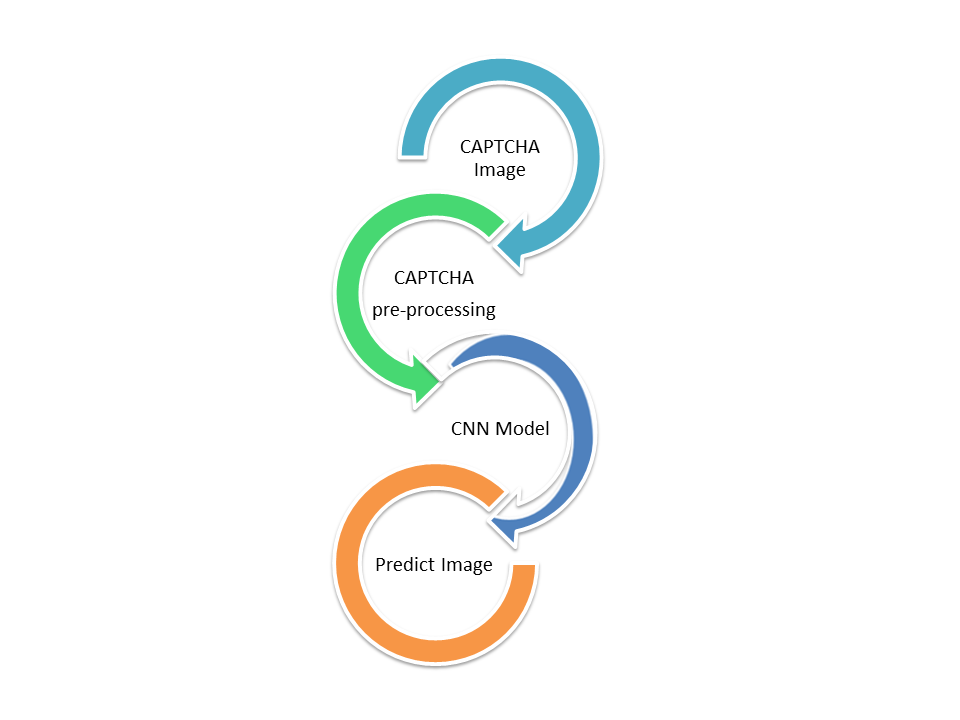


Figure 3 Working of the project

* 1. **Data Preprocessing:**

The dataset consists of images of size 50 height and 200 width. Each image consists of five letters consisting of small English alphabets (26) and digits (0-9 viz 10), thus making the total characters possible to 36. These images first need to be pre-processed before developing the model. For the purpose of training, the images in the dataset have a filename same as the CAPTCHA possessed by the image.

The images are first pre-processed by reading them in grayscale. This helps us to remove the noise to some extent as the images get invariant of the background color. The grayscale image consists of only one channel which makes it easier to further develop the model instead of using the three channels for RGB. Simultaneously, the file name is also stored in a string. From this filename, we drop the last four characters which are the “.png” characters containing the file extension, such that the remaining filename contains only the CAPTCHA. Now, if the length of this modified filename is less than six, it means that the CAPTCHA in the image does not have five characters and that particular image is not used further in the model.

Each grayscale image is then scaled and reshaped to the size: height- 50, width-200 and the number of channels as 1. Then we create an array of dimension 5\*36(5 denotes the number of characters in the CAPTCHA and 36 denotes all the possible characters) for each image containing all entries as zeroes. This array is used to store the character present at each position in the CAPTCHA. At each position, through filename we find which characters are present at each position of the CAPTCHA and update the corresponding location to one in this array. This array will thus be used for training the model.

Thus, after pre-processing we obtain all the grayscale images and the target array containing information regarding the characters present in each CAPTCHA image.

* 1. **Model development:**

The model developed for this CAPTCHA dataset uses Convolutional Neural Network. It consists of a total twenty four layers comprising the input layer, convolutional layers, max pooling layers, dense layers, flatten layers and dropout layers. The total numbers of parameters is 1,818,196 where 1,818,132 parameters are trainable and 64 are non-trainable parameters. A brief architecture of the layers is depicted in figure 3.

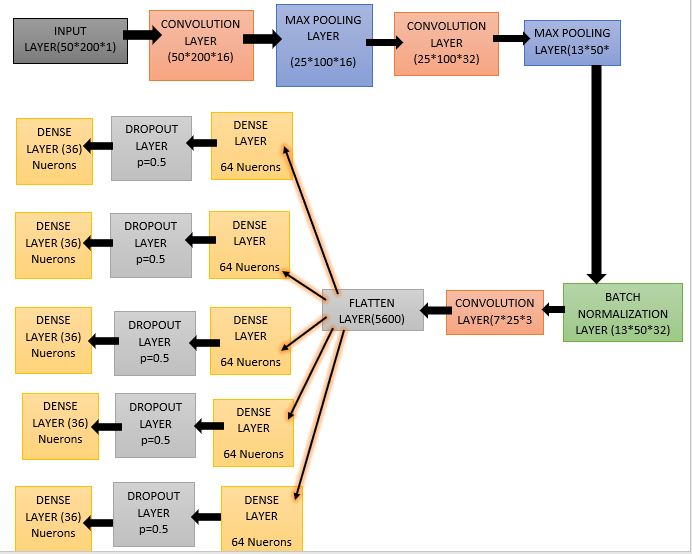


Figure 4 Architecture of the layers

The first layer is the input layer which takes the image of shape 50 height, 200 width and number of channels 1 as an input. The input layer is connected to the convolutional layer 1 which consists of 16 filters each of size (3,3) . The layer uses the stride of 1 and the padding is ‘same’ which means that padding is applied such that the output size and the input size is the same. The activation function deployed for this layer is ‘relu’. Relu is a rectified linear activation function which gives returns same as input if the input is positive else it returns zero. This convolutional layer 1 contains a total of 160 parameters. This layer transforms the shape from (50, 200, 1) to (50,200,16).

The convolutional layer 1 is connected to the max pooling layer 1. The padding used is ‘same’. This layer chooses the most prominent feature thus reducing the computational size to (25, 100,16). This also helps to reduce over fitting. Followed by the max pool layer 1 is again a convolutional layer. This convolutional layer now uses 32 filters instead of 16 where each filter size is (3,3). The padding used is ‘same’ and the activation function is ‘relu’. The parameters trained in this layer are 4640 and this layer changes the shape by increasing the number of channels and yields the output as (25, 100, 32). Now, this convolutional layer is again followed by the max pooling layer 2 which has the same hyper parameters as max pool layer 1 and it reduces the dimensions to (13, 50, 32). This layer is again followed by the convolutional layer 3 which has the same hyper parameters as convolutional layer 2 and it does not alter the shape. The numbers of parameters in this layer are 9248.

The next layer is the batch normalization layer. This is used to improve the stability of the model. This layer is used to normalize the activations of the convolutional layer 3 at each batch. This layer uses a transformation in such a way that the mean activation is maintained close to zero and the activation standard deviation is close to one. The numbers of parameters trained in this layer are 128 and the shape is not altered. Following this layer is again a max pooling layer which uses padding as ‘same’ and changes the output shape to (7, 25, 32).

After the max pooling layer, we have a flatten layer which converts the input from max pool layer to a long vector of 5600 dimensions. This is done so as the further neural network is easily processed and the back propagation is carried out easily. Further there are five dense layers in this neural network each of which is connected to the flatten layer. Each of these dense layers has the dimension 64 and deploys the activation function ‘relu’ to train a total of 358464 parameters each. Further to each of these dense layers is connected a dropout layer. The dropout layer drops 0.5 fraction of the nodes at the dense layer so that the model does not learn from the noise present in the dataset as well as the nodes probabilistically tries to learn from the inputs. Following the dropout layer is again a dense layer which uses the activation function ‘sigmoid’. The sigmoid function is also called a logistic function and it transforms the input to values between 0 and 1. The sigmoid function is a s-shaped curve such that as x approaches negative infinity, y approaches zero; as x approaches to infinity, y approaches one and at x equals to zero, y is 0.5. Each dense layer trains 2340 parameters and gives an output of dimension 36. All these layers and parameters are represented in tabular form in the figure 5.

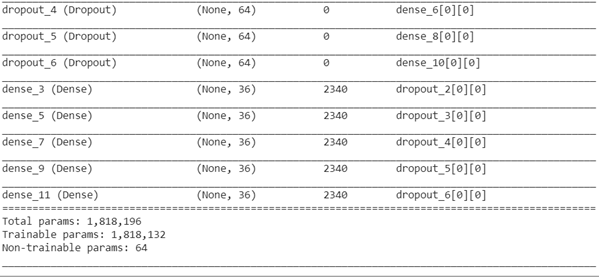
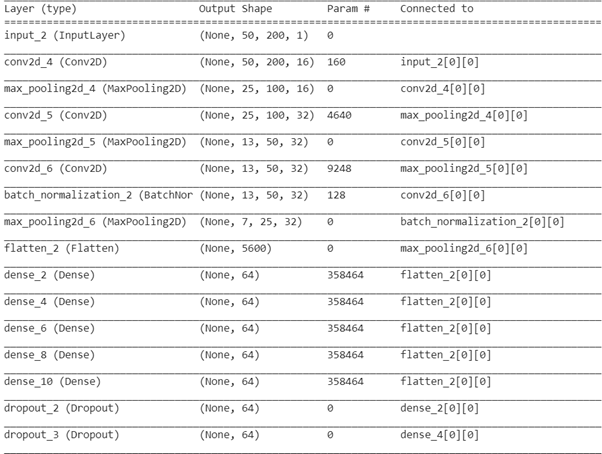


Figure 5

Thus, the model uses an image of dimension (50, 200, 1) as input and gets output from 5 layers each having dimension 36. The model uses the optimizer ‘Adam’ which is a stochastic gradient descent method that uses parameter learning rates based on the average of first moment and second moment of the gradients[11]. The model uses loss function as ‘categorical cross entropy’. Categorical cross entropy is a version of Binary cross entropy which determines the variation between the targeted distribution and normal distribution. Categorical cross entropy is used when one sample has various classes or labels as in this case the character set consists of 36 labels[13].The metrics used is ‘accuracy’.

The dataset consists of 1070 sample images out of which 970 images have been used for training and the remaining images for testing purpose. Further, for training the model, a validation split of 0.2 is used which splits the training set such that 80% of the training data is used for training and the remaining 20% for testing. The batch size used is 32 which denote the number of samples per gradient update and the number of epochs used is 60. The results of this training are discussed in further sections.

* 1. **Results and Discussion:**

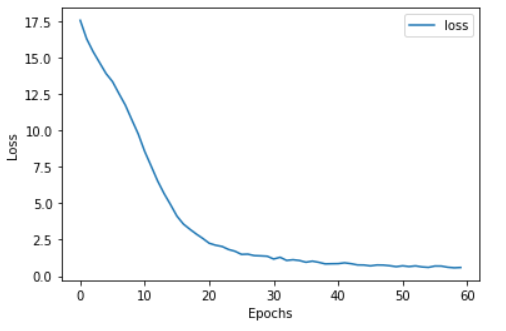
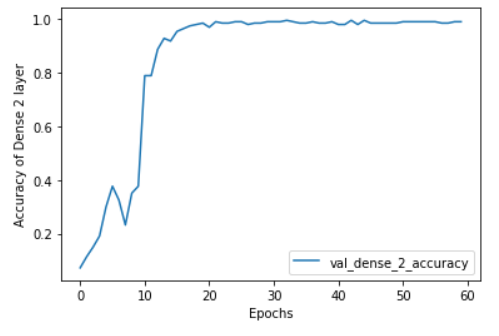
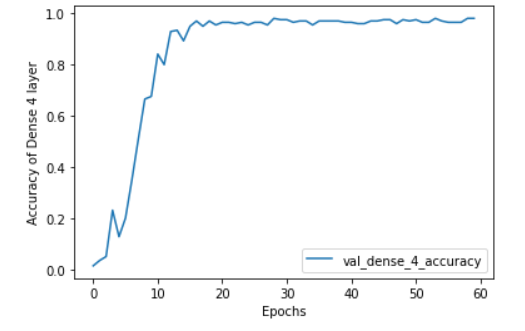
After training the above model for 60 epochs, the following graph was obtained for loss with respect to the number of epochs as shown in figure 6. We see that as the number of epoch’s increases, the loss decreases exponentially. The loss at the end of 60 epochs is 0.5932. The loss obtained on training set is 0.2391 while the loss on test set is 2.123.

Figure 6 Graph for epochs vs. loss

Next, we analyze how the accuracy obtained at each last dense layer varies with the number of epochs. The graph for accuracy of the output dense layers, namely dense layer 2,4,6,8 and 10 with respect to number of epochs is shown in figures 7,8,9,10,11. Thus we see that as the number of epochs increases, the accuracy of the layers improves and hence the system can predict the CAPTCHA more efficiently. The accuracy obtained after 60 epochs for dense layer 2 is 0.9897, dense layer 4 is 0.9794, dense layer 6 is 0.9227, dense layer 8 is 0.8969 and for dense layer 10 is 0.9278.

**Figure 7 Graph for epochs vs. accuracy for Dense 2 layer **



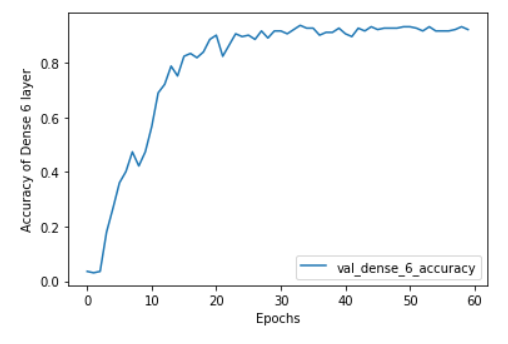
Figure 8 Graph for epochs vs. accuracy for Dense 4 layer

Figure 9 Graph for epochs vs. accuracy for Dense 6 layer

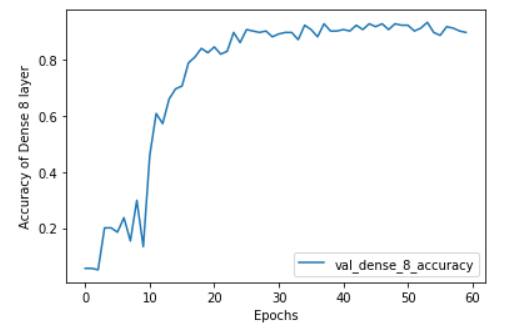
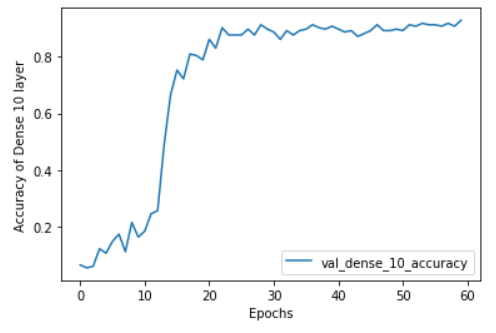
 Figure 10 Graph for epochs vs. accuracy for Dense 8 layer

Figure 11 Graph for epochs vs. accuracy for Dense 10 layer

Now, we predict the CAPTCHA. For predicting the CAPTCHA, we provide the image path and then the image is scaled. After that the model uses the image and then the output from the layers of model is mapped with the character set and the CAPTCHA is predicted. As a sample, an image from the dataset is passed to the model separately after renaming the file to some random name. The figure and the predicted CAPTCHA are shown in figures 12 and 13.

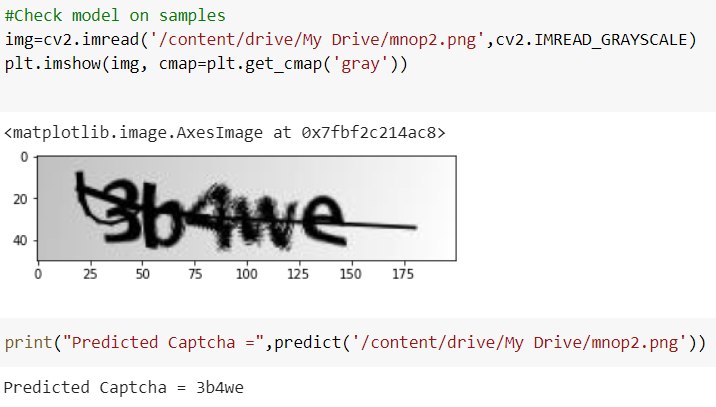


Figure 12

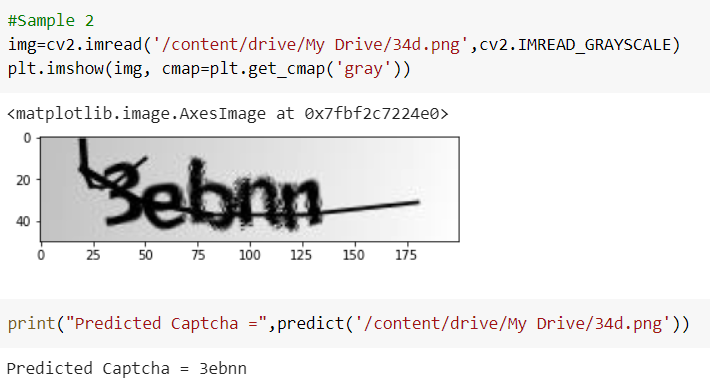


Figure 13

Thus, we see that our model predicts the CAPTCHA efficiently for small letters of English language and digits.

1. **Conclusion:**

CAPTCHA was designed to improve the security of the systems but deep learning algorithms defeated its very purpose. In this project, we used Convolutional Neural networks for CAPTCHA recognition. The dataset consisted of 1070 sample images and consisted of 5 lettered CAPTCHA. The model has been trained for CAPTCHA containing small English alphabets and digits, thus for a total of 36 characters. For training a purpose 970 samples have been utilized. The images in the dataset have been preprocessed and converted into grayscale for the further training. While training a convolutional neural network of twenty four layers has been developed consisting of input layer, convolutional layers, max pooling layers, flatten layer, dense layers and dropout layers. The model outputs five layers corresponding to each character of CAPTCHA and each has dimension 36 corresponding to the total number of characters possible. The model is trained using 60 epochs and it works well to predict any 5 lettered CAPTCHA containing small English alphabets and digits. The loss obtained after 60 epochs is 0.5932 and the accuracy of the output layers obtained is as dense layer 2 - 0.9897, dense layer 4 - 0.9794, dense layer 6 - 0.9227, dense layer 8 - 0.8969 and for dense layer 10 - 0.9278. The loss on training set is 0.2391 while the loss on test set is 2.123. The future scope of this work lies to expand this CAPTCHA recognition system for larger and more noisy CAPTCHA containing all the symbols possible.

**REFERENCES:**

1. Pulli, Kari; Baksheev, Anatoly; Kornyakov, Kirill; Eruhimov, Victor (1 April 2012). ["Realtime Computer Vision with OpenCV"](http://dl.acm.org/citation.cfm?id=2206309). *Queue*: 40:40–40:56. [doi](https://en.wikipedia.org/wiki/Doi_(identifier)):[10.1145/2181796.2206309](https://doi.org/10.1145%2F2181796.2206309)(inactive 2020-05-21)
2. Jun Chen,1,2 Xiangyang Luo,1 Yanqing Guo,3 Yi Zhang,1 and Daofu Gong1, A Survey on Breaking Technique of Text-Based CAPTCHA, Hindawi Security and Communication Networks Volume 2017, Article ID 6898617, 15 pages.
3. Fabian Stark, Caner Hazırba¸s, Rudolph Triebel, and Daniel Cremers, CAPTCHA Recognition with Active Deep Learning, Technical University of Munich, Germany .
4. ["R interface to Keras"](https://keras.rstudio.com/). *keras.rstudio.com*.
5. <https://vinodsblog.com/2018/10/15/everything-you-need-to-know-about-convolutional-neural-networks/>
6. Artificial Neural Networks Technology - A DACS State-of-the-Art Report, by Dave Anderson and George McNeill.
7. CAPTCHA recognition based on deep convolutional neural network Jing Wang, Jiaohua Qin \*, Xuyu Xiang, Yun Tan and Nan PanCollege of Computer Science and Information Technology, Central South University of Forestry andTechnology, 498 shaoshan S Rd, Changsha,410004,China.DOI:10.3934/mbe.2019292
8. Boukari Souley and Hauwa Abubakar, A CAPTCHA – Based Intrusion Detection Model,International Journal of Software Engineering & Applications (IJSEA), Vol.9, No.1, January 2018
9. L. Zhang, S. W. Huang, Z. X. Shi, et al., CAPTCHA recognition method based on LSTM RNN, Pattern Recogn., 1 (2011), 40–47.
10. Deep Learning Model for Breaking Text Based CAPTCHAs Sustainable Humanosphere ISSN: 1880 - 6503, | May 2020 Volume: 16 Issue: 2
11. <https://keras.io/api/optimizers/adam/>
12. CAPTCHA Breaking with Deep Learning: Nathan Zhao , Yi Liu , Yijun Jiang
13. <https://keras.io/api/losses/probabilistic_losses/#categorical_crossentropy-function>
14. Y. H. Shen, R. G. Ji and D. L. Cao, Hacking Chinese touclick CAPTCHA by multiscale corner struc-ture model with fast pattern matching, Proceedings of the ACM International Conference on Multimedia, (2014), 853–856.
15. https://www.researchgate.net/publication/248380891\_CAPTCHA\_dataset