ANKARA UNIVERSITY

DEPARTMENT OF COMPUTER ENGINEERING

COM3551-B ARTIFICIAL INTELLIGENCE PROJECT REPORT

INSTRUCTOR : DOÇ. DR. MEHMET SERDAR GÜZEL

# CAPTCHA BYPASSING USING DEEP NEURAL NETWORK

| | |
|---|---|
| SÜNBÜL KÜBRA YAZICI | BERKAY ÇELEBİ |
| 17290131 | 17290096 |
| ALPER TONGA ATALAYIN | MELİSA YILDIRIM |
| 17290082 | 17290133 |

2021

ANKARA

## CONTENTS

## 1. WHAT IS CAPTCHA?

CAPTCHA (a contrived acronym for "Completely Automated Public Turing test to tell Computers and Humans Apart") is a type of challenge response test used in computing to determine whether or not the user is human. General form of CAPTCHA requires someone to correctly evaluate and enter a sequence of letters or numbers perceptible in a distorted image displayed on their screen (Figure-1.1). Because the test is administered by a computer, in contrast to the standard Turing test that is administered by a human, a CAPTCHA is sometimes described as a reverse Turing test.

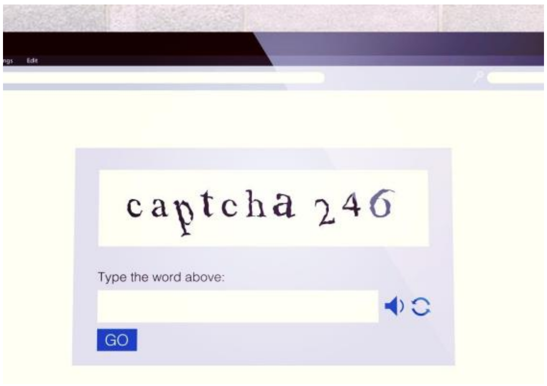


Figure-1.1 (An example of CAPTCHA)

### 1.1. WHY CAPTCHA BYPASS?

An attacker can create a bot to bypass the captcha and automate the tasks to send unlimited requests to multiple URLs or lists with random/fake users, emails, IP address. For spamming or evil purposes (collect data, analyze traffic behaviors, etc). An attacker can perform rate limiting or can use bots. There are several methods that are using for bypassing CAPTCHAs.

### 1.2. CAPTCHA BYPASSING METHODs

- It is an easy method to check for bypassing captcha just by changing the “request method of your request” and removing the captcha parameter.
- Sometimes you found that parameters are passing using JSON data. You can first convert it in a simple post request and try or combine it with the 1st method.
- Using extra headers for rate limiting.
- OCR(Optical Character Recognition) enabled bots. This particular approach solves CAPTCHAs automatically using Optical Character Recognition (OCR) technique. Tools like Ocrad, tesseract solve CAPTCHAs but with very low accuracy.
- Online CAPTCHA-solving services — The service has human workers who are available online constantly to solve CAPTCHAs. When you send your CAPTCHA solving request, the service forwards it to the solvers who break it and return the solutions.
- Using artificial intelligence models.
    - Most of the methods are used for specific vulnerabilities and misconfigurations, however artificial intelligence models can train for any types of CAPTCHAs.

### 1.3. CAPTCHA BYPASSING BY USING ARTIFICIAL INTELLIGENCE

AI has been part of our imaginations and simmering in research labs since a handful of computer scientists rallied around the term at the Dartmouth Conferences in 1956 and birthed the field of AI. In the decades since, AI has alternately been heralded as the key to our civilization’s brightest future, and tossed on technology’s trash heap as a harebrained notion of over-reaching propellerheads. Frankly, until 2012, it was a bit of both.

Over the past few years AI has exploded, and especially since 2015. Much of that has to do with the wide availability of GPUs that make parallel processing ever faster, cheaper, and more powerful. It also has to do with the simultaneous one-two punch of practically infinite storage and a flood of data of every stripe (that whole Big Data movement) – images, text, transactions, mapping data, you name it.

## Machine Learning - An Approach to Achieve Artificial Intelligence

Machine learning at its most basic is the practice of using algorithms to parse data, learn from it, and then make a determination or prediction about something in the world. So rather than hand-coding software routines with a specific set of instructions to accomplish a particular task, the machine is “trained” using large amounts of data and algorithms that give it the ability to learn how to perform the task.

Machine learning came directly from minds of the early AI crowd, and the algorithmic approaches over the years included decision tree learning, inductive logic programming. clustering, reinforcement learning, and Bayesian networks among others. As we know, none achieved the ultimate goal of General AI, and even Narrow AI was mostly out of reach with early machine learning approaches.

## Deep Learning - A Technique for Implementing Machine Learning

Another algorithmic approach from the early machine-learning crowd, artificial neural networks, came and mostly went over the decades. Neural networks are inspired by our understanding of the biology of our brains all those interconnections between the neurons (Figure-2.1). But, unlike a biological brain where any neuron can connect to any other neuron within a certain physical distance, these artificial neural networks have discrete layers, connections, and directions of data propagation. Each neuron assigns a weighting toits input how correct or incorrect it is relative to the task being performed (Figure-2.2). The final output is then determined by the total of those weightings. It comes up with a "probability vector," really a highly educated guess, based on the weighting. Chances are very good that as the network is getting tuned or "trained" it's coming up with wrong answers a lot. What it needs is training. It needs to see hundreds of thousands, even millions of images, until the weightings of the neuron inputs are tuned so precisely that it gets the answer right practically every time It's at that point that the neural network has taught itself your mother's face in the case of Facebook; or a cat, which is what Andrew Ng did in 2012 at Google.

Ng's breakthrough was to take these neural networks, and essentially make them huge, increase the layers and the neurons, and then run massive amounts of data through the system to train it. In Ng's case it was images from 10 million YouTube videos. Ng put the "deep" in deep learning, which describes all the layers in these neural networks.

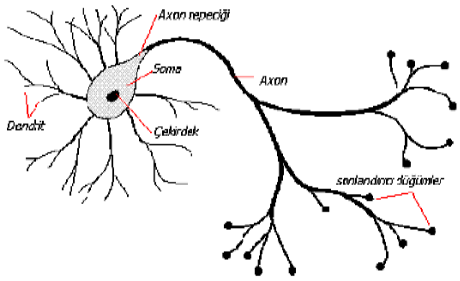


Figure-2.1(Typical Nerve Cell)

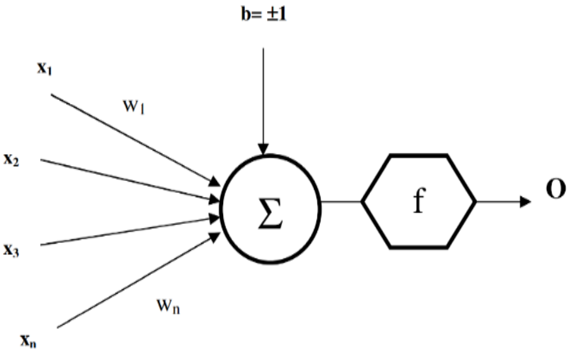


Figure-2.2(Basic Artificial Neural Network Cell)

## 1.4.WHY THE CAPTCHAs ARE UNSECURED?

The first major problem with CAPTCHA is that they are often thought of as a security measure, when in fact they are not. A CAPTCHA is a crude (automated) way of telling humans and computers apart. It does not, and cannot, test whether a user can be trusted or not. If you are using CAPTCHA for security, you are working on the false assumption that humans can be trusted while computers (bots) cannot.

So, the first basic problem of the CAPTCHA is that it is too often thought of as a security measure when in fact it is not. Thinking of CAPTCHA as a security measure is equivalent to thinking of all humans as trustworthy.

The second major problem with CAPTCHAs is that they are relatively easy to exploit. While a single human can only look at a certain number of images per hour, a number of humans, with a lot of time on their hands, can look at thousands of them per hour. And if the internet has taught us anything, it’s that there are a lot of humans on the internet with a lot of time on their hands. So, while a CAPTCHA will slow one human down, it won’t slow hundreds of humans down.

The third major problem with a CAPTCHA also harms usability. It takes time and effort for even a person of good eyesight to pass a CAPTCHA. It is a hassle and an annoyance.

## 2. WHAT IS OUR MODEL?

In this project our problem is that can we bypass simple CAPTCHAs with high accuracy.As we researched simple captchas,we figured out that there are still millions of web pages that are using simple captchas. For example in this project we referenced one of them "Really Simple Captcha" which is used in wordpress websites over 1,5 million times.

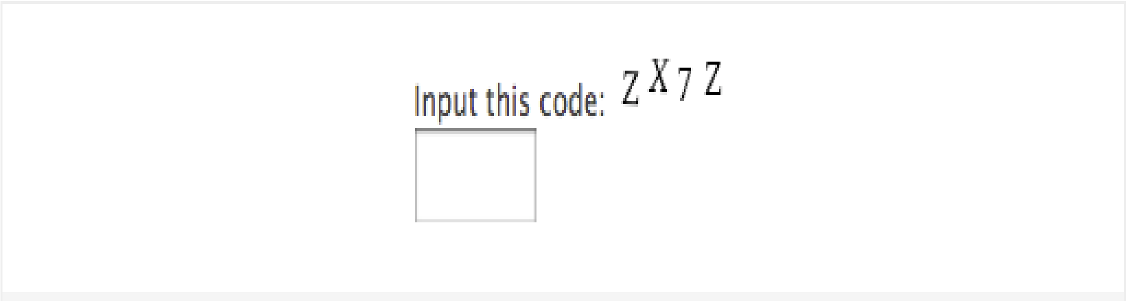


Figure-3.1(An example of “Really Simple Captcha”)

Really simple captcha basically uses a font and creates a ".png" file without customizing shapes of characters.

By using this reference, we created our CAPTCHA generator which also modifying shapes of characters therefore this CAPTCHA had been more complex than "Really Simple Captcha" (Figure-3.2).

Figure-3.2

## 2.1. CREATING DATASET

To train our model we created 10.000 different ".png" files to provide enough data. Each element of dataset consists of 20 numbers of characters which are:

'A','B','C','D','E','F','G','H','K','P','0','1','2','3','4','5','6','7','8','9'

To preserve confliction, like similarity between characters "l" and "1" or "0" and "O", we removed some letters.

## 2.2. PREPARING DATASET FOR TRAINING THE MODEL

Each file in our dataset is named with sequences of characters in that file towards left to right like Figure-3.3.

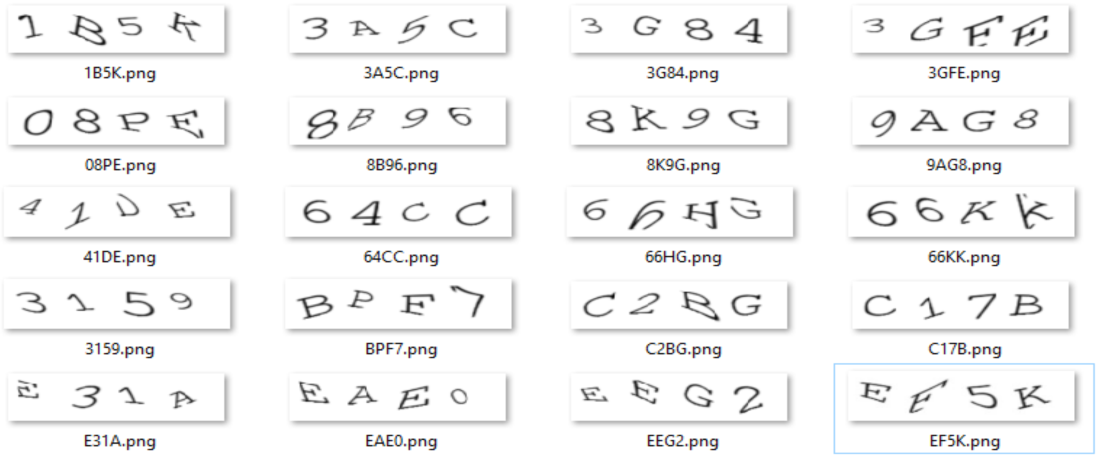


Figure-4.1

The main problem is that training the model without separating images is challenging. So that we investigated structure of files and we saw that each character is separate from each other. Then we realized that we could create a new dataset, which only have one character long, derived from our dataset.

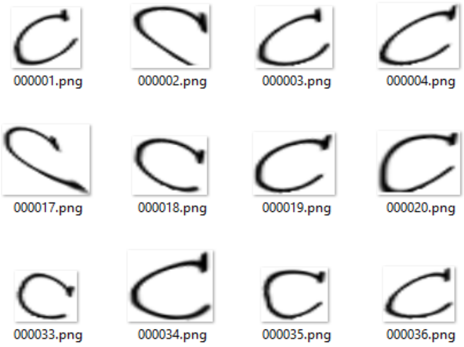


Figure-4.2

To create this dataset firstly we used findContours() function of python module named "OpenCV". This function basically finds similar colors with respect to density of each pixels and we cut every separate character by this method.

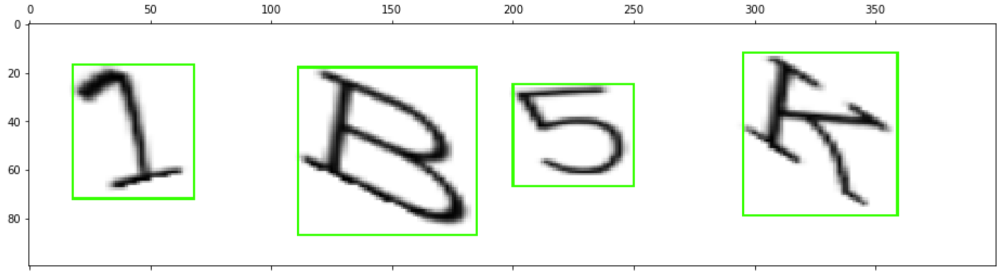
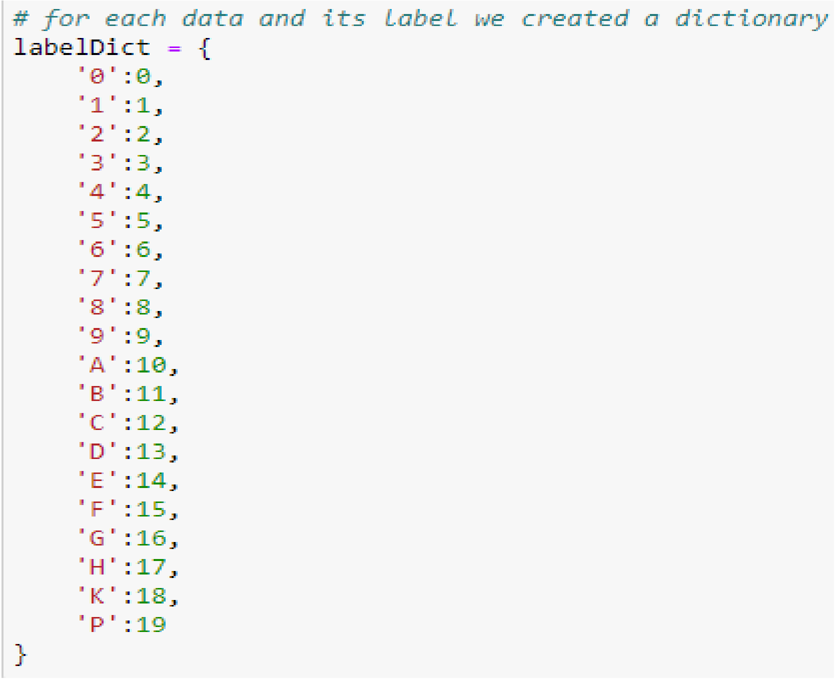


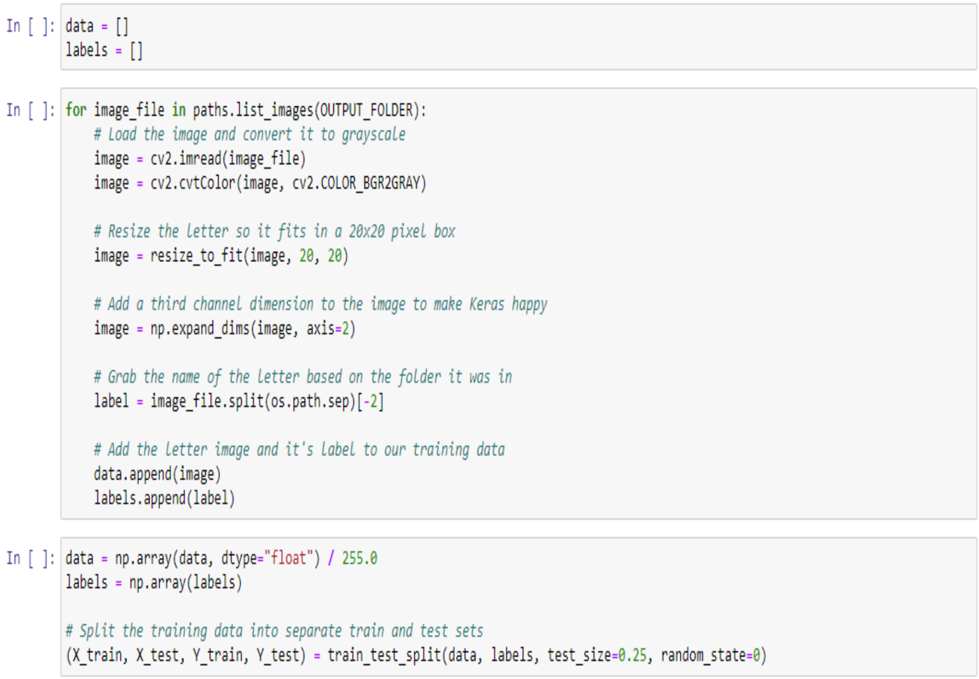
Figure-4.3

To represent each character numerically we created a dictionary.



```
# for each data and its label we created a dictionary
labelDict = {
        '0':0,
        '1':1,
        '2':2,
        '3':3,
        '4':4,
        '5':5,
        '6':6,
        '7':7,
        '8':8,
        '9':9,
        'A':10,
        'B':11,
        'C':12,
        'D':13,
        'E':14,
        'F':15,
        'G':16,
        'H':17,
        'K':18,
        'P':19
}
```

Figure-4.4



```
In [ ]: data = []
        labels = []
```

```
In [ ]: for image_file in paths.list_images(OUTPUT_FOLDER):
            # Load the image and convert it to grayscale
            image = cv2.imread(image_file)
            image = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)

            # Resize the letter so it fits in a 20x20 pixel box
            image = resize_to_fit(image, 20, 20)

            # Add a third channel dimension to the image to make Keras happy
            image = np.expand_dims(image, axis=2)

            # Grab the name of the letter based on the folder it was in
            label = image_file.split(os.path.sep)[-2]

            # Add the letter image and it's label to our training data
            data.append(image)
            labels.append(label)
```

```
In [ ]: data = np.array(data, dtype="float") / 255.0
        labels = np.array(labels)

        # Split the training data into separate train and test sets
        (X_train, X_test, Y_train, Y_test) = train_test_split(data, labels, test_size=0.25, random_state=0)
```

Figure-4.5

We loaded all of the files to data and label lists and converted them to pandas array. Also, we resized images to fixed size 20x20 pixels (Figure-3.8).

After that we have created our training and test datasets by 3 to 1 rate.

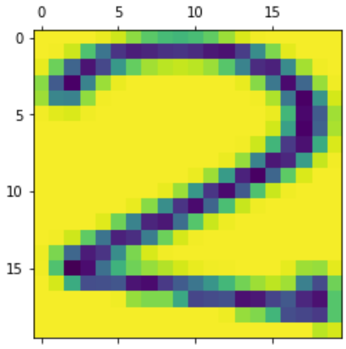
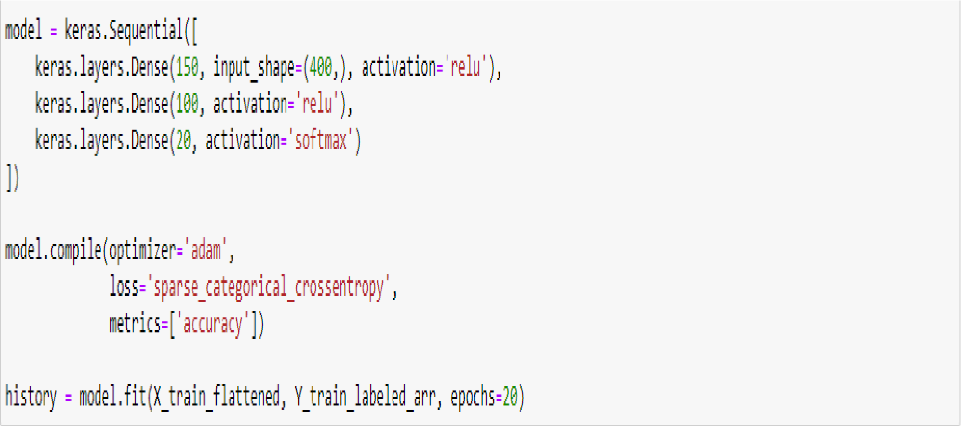


Figure-4.6

Because of we transformed every picture into 2D gray scale arrays our model will see them shown in above.

## 2.3. MODELING NEURAL NETWORK

The best neural network approach for this type of classification problems is linear neural network. We used keras library for model.



```
model = keras.Sequential([
    keras.layers.Dense(150, input_shape=(400,), activation='relu'),
    keras.layers.Dense(100, activation='relu'),
    keras.layers.Dense(20, activation='softmax')
])

model.compile(optimizer='adam',
              loss='sparse_categorical_crossentropy',
              metrics=['accuracy'])

history = model.fit(X_train_flattened, Y_train_labeled_arr, epochs=20)
```

Figure-5.1

In this model we have input layer which is responsible for each pixel then in hidden layer the model is splitting into meaningful shapes like curves, lines etc. In output layer 20 neurons are responsible for classifying.

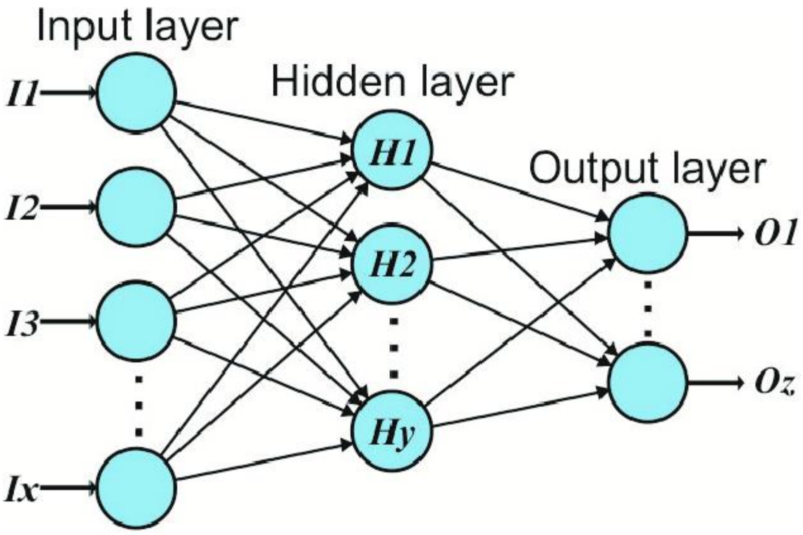
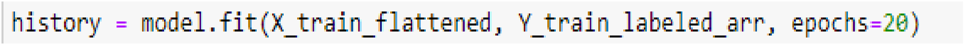


Figure-5.2

## 2.4. TRAINING AND TESTING THE MODEL

To train model we gave the X_train_flattened array with 20 epoches to the model.



```
history = model.fit(X_train_flattened, Y_train_labeled_arr, epochs=20)
```

After training we obtained 0.97 accuracy as a result.

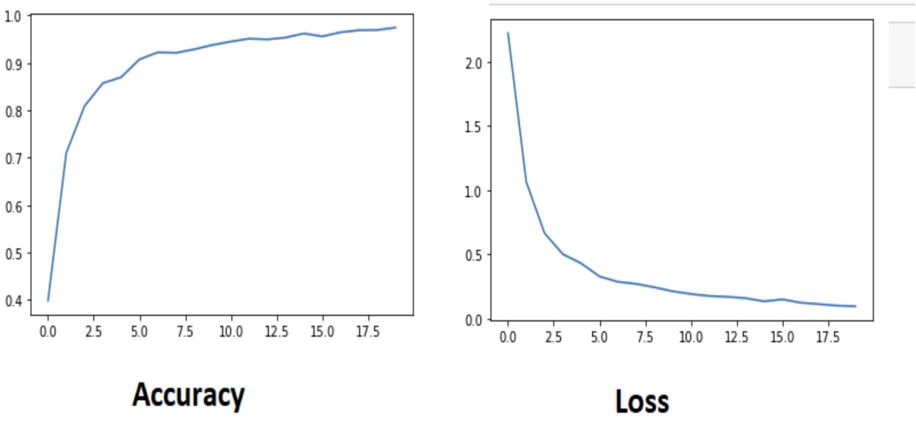
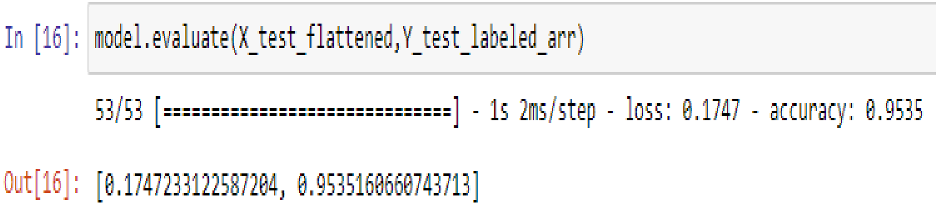


Figure-6.1

When we tested our model with X_test_flattened we obtained 0.95 accuracy. This is enough rate for classifying characters of CAPTCHA.

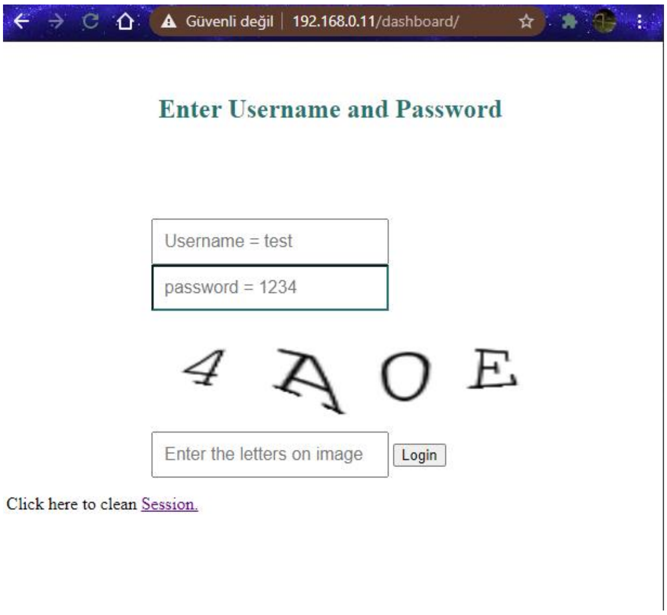


```
In [16]: model.evaluate(X_test_flattened,Y_test_labeled_arr)

         53/53 [==============================] - 1s 2ms/step - loss: 0.1747 - accuracy: 0.9535

Out[16]: [0.1747233122587204, 0.9535160660743713]
```

Figure-6.2

## 2.5. TESTING THE MODEL WITH TEST PICTURES

To simulate real world case, we created a test webpage.



Enter Username and Password

Username = test

password = 1234

4 A O E

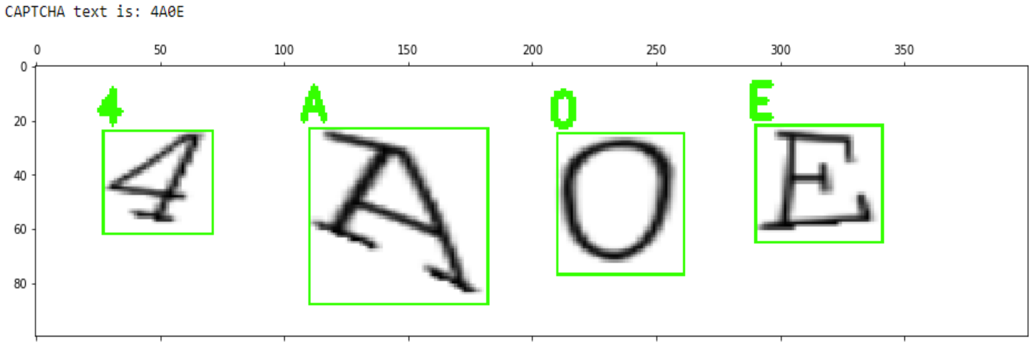
Enter the letters on image

Login

Click here to clean Session.

Figure-7.1

Also, to identify the characters we have written a script. We copy the url of the image on the page and give the link to our script and it gives us the result shown in below.



```
CAPTCHA text is: 4A0E
```

Figure-7.2

The script basically does the same things we have done to split letters from image like before ones also merge all identifications together.

## 3. HOW TO GENERATE MORE SECURED CAPTCHAS?

Generating a secured CAPTCHA is an endless war between machines and humans. If we generate a CAPTCHA that any AI cannot bypass, then we can say that any AI will never act like a human and this is a philosophical problem. However, using updated CAPTCHAs like reCAPTCHA can be used against to attacks.

To make a CAPTCHA more secured only way is making the CAPTCHAS more complex but as we mentioned before it will not never totally safe.

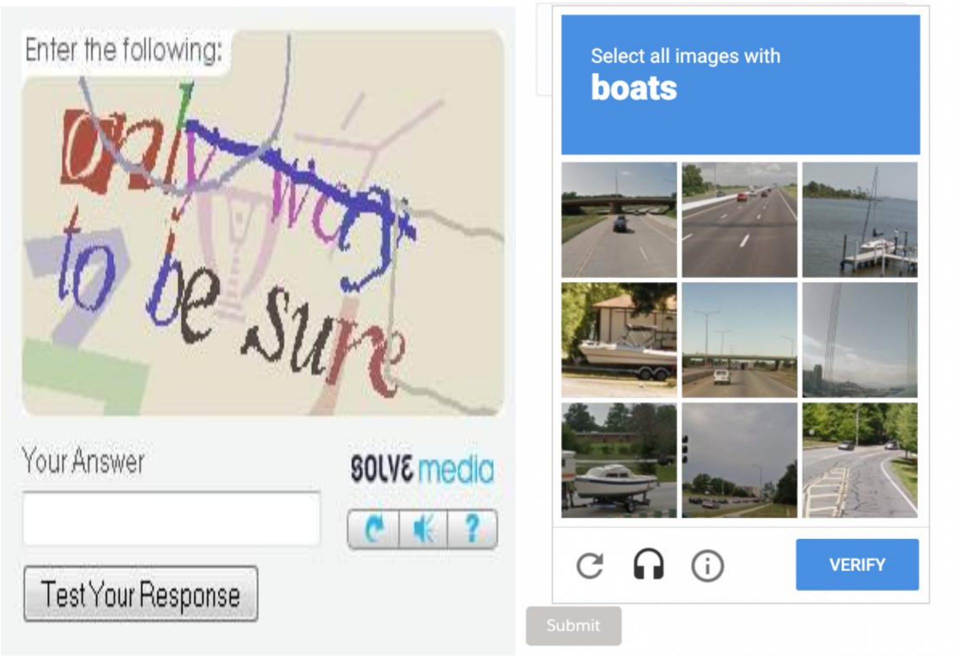


Figure-8.1

## 4. RESULT

Consequently, even in this simple training model we have achieved almost 91% accuracy. If we think about there are lots of websites that are using these types of simple CAPTCHAs, with more complicated neural networks like CNN, the accuracy rate may achieve almost 99%. This may cause pretty much security problems.

## 5. GITHUB LINK OF OUR RESEARCH

**https://github.com/berkaycelebi00/simplecapthcabypassAI/**

## 6. YOUTUBE VIDEO LINK

https://youtu.be/Q4BGdKWQJfU

**RESOURCES:**

- **https://www.researchgate.net/figure/Feed-forward-neural-network-a-Architecture-b-Nonlinear-model-of-neuron_fig2_318740391**

- **https://blogs.nvidia.com/blog/2016/07/29/whats-difference-artificial-intelligence-machine-learning-deep-learning-ai/**

- **https://towardsdatascience.com/solving-captchas-machine-learning-vs-online-services-3596ad6f0137**

- **https://medium.com/@honeyakshat999/captcha-bypass-techniques-f768521516b2**

- **https://www.dictionary.com/e/captcha/**

- https://medium.com/@ageitgey/how-to-break-a-captcha-system-in-15-minutes-with-machine-learning-dbebb035a710
- https://en.wikipedia.org/wiki/CAPTCHA
- **Deep-CAPTCHA: a deep learning based**

  **CAPTCHA solver for vulnerability assessment**

  **Zahra Noury, Mahdi Rezaei y**

  **Faculty of Computer and Electrical Engineering, Qazvin Azad University**

  **Faculty of Environment, Institute for Transport Studies, The University of Leeds**
- **CAPTCHA Breaking with Deep Learning - CS 229 Final Project, Autumn 2017 Nathan Zhao Yi Liu Yijun Jiang**