I AM A ROBOT: SOLVING CAPTCHA WITH DEEP LEARNING

A PREPRINT

Jonathon Howland
Tippie College of Business Analytics
University of Iowa
Iowa City, IA
jonathon-howland@uiowa.edu

April 1, 2021

ABSTRACT

1 Introduction

We've all seen them: the little boxes you had to check on websites stating that "I'm not a robot". Then, you had to look at some jumbled, scribbled-on, multi-colored characters and type what they said. And, honestly, I failed sometimes. Fortunately, we don't see those very often anymore, and there's a reason: the robots are smart enough to solve them. Which is impressive, considering the fact that I'm a human and I have trouble with them sometimes.

Those sets of characters are known as CAPTCHAs. CAPTCHA stands for "Completely Automated Public Turing test to tell Computers and Humans Apart". It's a bit contrived, but it gets the point across. For those who are unfamiliar with it, a Turing Test is a general term for a test that shows a computer has intelligence comparable to a human's in a given task. These were designed to be difficult for a machine learning model to solve. In fact, that is the entire point of their existence. [1]

Over time, however, we've begun seeing fewer CAPTCHAs. Instead, they are being replaced with image recognition tests. The reason is that with the advent of deep learning, it is possible to train a model that can decipher these CAPTCHAs. In general, it is still a very difficult problem. Characters can be smashed together to appear as one instead of two or more. Lines and patterns can look like extra characters. Some even have multiple colors within each character.

For this particular project, we've made the problem a bit easier. We are only looking at CAPTCHAs with exactly five characters. The images are also all from a similar type of CAPTCHA. They all have five distorted, colored characters on a distorted background. All of the images have markings in across them, similar to if somebody took a pen and drew some scribbles on them (a couple examples can be seen in the "Data" section).

The fact that we know each image has exactly five characters is probably the most helpful thing for us. Our goal is to go through the image and identify each character sequentially. Knowing that there are always five helps in that if there are, for example, two "u"s squished together, we know that they must be not be a "w" (though we may not know if they are two "u"s, a "u" and a "v", etc.).

Although we have yet to cover it in class, this problem will utilize a recurrent neural network (RNN). This will allow us to sequentially identify the characters in the CAPTCHA, making it an excellent choice for this project. Once the characters have been identified, they can be compared to the ground truth. Each CAPTCHA is one image, and the labels are five character strings that we can compare to our results from the neural network.

2 Problem Definition

To be precise, the goal is to identify the five characters contained in a CAPTCHA image, which will then be compared to the five characters contained in the label to determine accuracy. The CAPTCHA images and labels are the only inputs for this project, and the predicted labels are the outputs.

Each CAPTCHA image is a 150x40 color image containing five characters, as well as various other markings. In the event that an image is not 150 by 40 pixel, it will be resized to those dimensions. In order for the machine learning process to run more smoothly, we will also normalize all the RGB values for the pixels to the range [-1, 1], as opposed to their original ranges of [0, 255].

The true labels can be obtained from the file names, which are all in the format of <label>.jpg, where <label> is the five-character string containing the true values of the CAPTCHA. Due to the massive size of the dataset, I will not be manually confirming the accuracy of the labels. They were pre-defined by the original creator of this data, Parsa Sam [3].

3 Data

I say that the dataset is massive, although not compared to the likes of ImageNet. The data I am using is comprised of 113,062 CAPTCHA images and labels that were generated by the Captcha Library for PHP, which was created by Grégoire Passault [2]. While this isn't a huge number as far as deep learning goes, it is still far more than I am able to go through by hand.

As I have stated previously, each image contains 5 characters and some markings in the form of irregular curves across the image. From a cursory glance through the files, most, if not all, of the images have a similar format. So, I have selected two examples to show the basic idea of this particular CAPTCHA (Figures 1 and 2).

The images and file names can be fed into my code to extract the corresponding label for each image. Aside from that, most of the cleaning and parsing has already been done for me, which makes this project a lot easier.

I got this data from Kaggle.com, and Parsa Sam, the user who has it posted, has done a very nice job of getting the data into an easy format to feed into TensorFlow [3]. To summarize, the data is in the form of approximately 113,000 JPG image files, with each file's name containing the label in the form <label>.jpg.

Figure 1: The CAPTCHA image with file name "1a1SZ.jpg". The corresponding label is therefore "1a1SZ", which seems to be the characters this CAPTCHA contains.

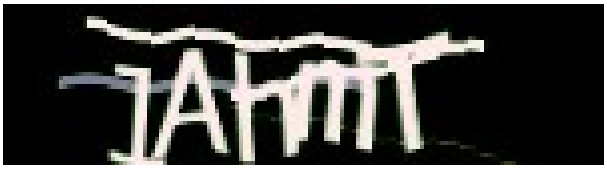

Figure 2: The CAPTCHA image with the label "1AhmT". Again, the label seems to match the characters shown in the image. Though, honestly, I have trouble with this one even as a human.

4 Other sections

References

[1] *CAPTCHA - Wikipedia*. 2021. URL: https://en.wikipedia.org/wiki/CAPTCHA.

[2] Grégoire Passault. *Captcha Library for PHP*. 2020. URL: https://github.com/Gregwar/Captcha.

[3] Parsa Sam. *CAPTCHA Dataset*. 2021. URL: https://www.kaggle.com/parsasam/captcha-dataset.