CAPTCHA Breaker

Moise Valentin

University of Bucharest

Introduction

CAPTCHAs (Completely Automated Public Turing test to tell Computers and Humans Apart) are used to determine if a user is a human or not. They are consisting of text or image distortions which can still be recognized by humans, but for a computer it represents a difficult task.

Contribution:

- Custom dataset, easy to modify with a large amount of images grouped by solving difficulty.
- Solving single letter Captchas with a CNN approach.

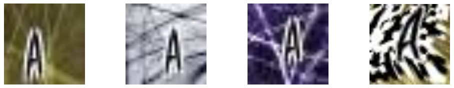
Further development:

- Recurrent approach.
- Solve Captchas of variable length.

Dataset

The dataset used was created with PyCaptcha. For any given length, 20000 easy, medium and hard to solve images containing letters with random noise have been generated.

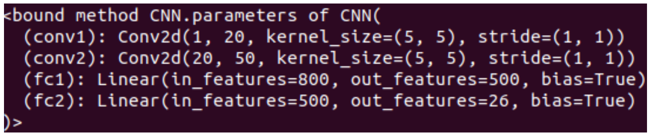
Single letter generated images, from easy to hard difficulty, 40x40 pixels:



4 letter generated image with easy difficulty, 150x50 pixels:

Model

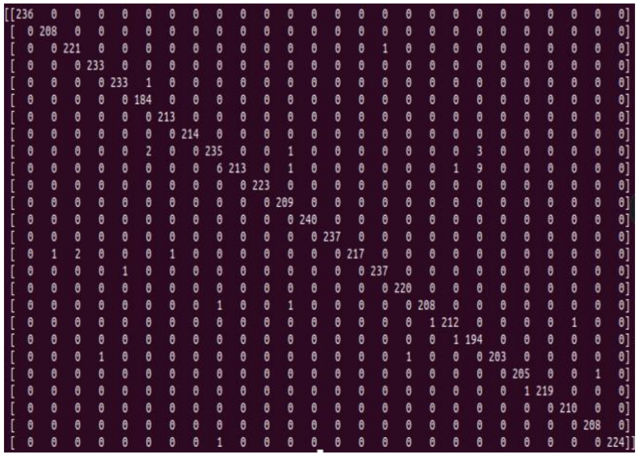
The single-letter captcha recognition was made using a Convolutional Neural Network, with the following architecture:



```
<bound method CNN.parameters of CNN(
  (conv1): Conv2d(1, 20, kernel_size=(5, 5), stride=(1, 1))
  (conv2): Conv2d(20, 50, kernel_size=(5, 5), stride=(1, 1))
  (fc1): Linear(in_features=800, out_features=500, bias=True)
  (fc2): Linear(in_features=500, out_features=26, bias=True)
)>
```

Results

On the easy images the model obtained an accuracy of 99%, and an average loss of 0.02



```
[[236   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0]
 [  0 208   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0]
 [  0   0 221   0   0   0   0   0   0   0   0   0   0   0   0   1   0   0   0   0   0   0   0   0   0   0]
 [  0   0   0 233   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0]
 [  0   0   0   0 233   1   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0]
 [  0   0   0   0   0 184   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0]
 [  0   0   0   0   0   0 213   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0]
 [  0   0   0   0   0   0   0 214   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0]
 [  0   0   0   0   0   2   0   0 235   0   0   1   0   0   0   0   0   0   0   3   0   0   0   0   0   0]
 [  0   0   0   0   0   0   0   0   6 213   0   1   0   0   0   0   0   0   1   9   0   0   0   0   0   0]
 [  0   0   0   0   0   0   0   0   0   0 223   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0]
 [  0   0   0   0   0   0   0   0   0   0   0 209   0   0   0   0   0   0   0   0   0   0   0   0   0   0]
 [  0   0   0   0   0   0   0   0   0   0   0   0 240   0   0   0   0   0   0   0   0   0   0   0   0   0]
 [  0   0   0   0   0   0   0   0   0   0   0   0   0 237   0   0   0   0   0   0   0   0   0   0   0   0]
 [  0   1   2   0   0   0   1   0   0   0   0   0   0   0 217   0   0   0   0   0   0   0   0   0   0   0]
 [  0   0   0   0   1   0   0   0   0   0   0   0   0   0   0 237   0   0   0   0   0   0   0   0   0   0]
 [  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0 220   0   0   0   0   0   0   0   0   0]
 [  0   0   0   0   0   0   0   0   1   0   0   1   0   0   0   0   0 208   0   0   0   0   0   0   0   0]
 [  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   1 212   0   0   0   0   1   0   0]
 [  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   1 194   0   0   0   0   0   0]
 [  0   0   0   1   0   0   0   0   0   0   0   0   0   0   0   0   1   0   0   0 203   0   0   0   0   0]
 [  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0 205   0   0   1   0]
 [  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   1 219   0   0   0]
 [  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0 210   0   0]
 [  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0 208   0]
 [  0   0   0   0   0   0   0   0   1   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0 224]]
```

On the hardest dataset the accuracy is around 88%, and with mixed images 90%.

The most errors were made when the image contained an "I" or "J", or any other similar letters combinations. These particular images were a difficult task even for humans because of the high noise added in the background.

References

PyCaptcha -
https://github.com/lerouxb/PyCAPTCHA