

Forum:[Hack Forums](#)

Thread: [\[TUT\] How to make a deep learning based CAPTCHA cracker](#)

kh40tika - September 6, 2017, 5:41 am

I'm not going to release the code here due to the harm it would potentially done. Also I wouldn't go into basics for machine learning / deep learning. I'm pretty sure there are tons tuts online. This tut is for people with intermediate deep learning / machine learning knowledge. For the missing details, figure the shit out on you own.

[How it works]

Most CAPTCHA contains an image of character string. Our method targets that. We build a CNN + LSTM hybrid deep neural network. It will convert a horizontal image into a time series of probability distribution over text characters. Then we will perform Connectionist Temporal Classification (CTC) on this pdf time series to obtain a string sequence. If you know how end-to-end speech recognition works, you should know what I'm talking about.

[Tools]

You will need a dataset, a deep learning framework (I use Tensorflow), and a decent NVIDIA GPU.

[Create the dataset]

The best way to create the dataset is to generate it on the fly. You should setup a proper pipeline, use CPU to generate images, then use GPU to train your nets.

If you target a specific CAPTCHA breed, you can get its API to generate the images. Otherwise you might want to write your own image generator. This would include font selection, distortion, and adding noise.

We will always feed images with same height to neural net, say, 48 px. This is due to model constraint. In runtime, if an image don't have the right height, we just scale it.

The dataset should have <image, string> pair for each data point, acting as input and label. This is supervised learning.

[Build the model]

So the model has two parts. One is feed forward ConvNet that acts like an 48 x 48 character recognizer. The ConvNet slides through the horizontal image to generate a time series. Then add a few layers of bidirectional LSTMs to reduce the effect of noise and overlapping characters.

I won't put detailed architecture here. But mostly, I use Wide-Residual-Network for ConvNet part, you can easily google for its arXiv paper. Then I apply two layers of BLSTMs. Feel free to experiment with other model settings. Finally I apply a softmax layer + differentiable CTC loss. I use BAIDU's warp CTC which runs on GPU.

[Training]

This part is no brainer. The model is end-to-end. Feed the data into your GPU and wait for a day or

two. If you use generator for dataset, you won't need a validation set or any kind of regularization. But do keep an eye for NaN or bad initialization.

[Result]

I was able to get ~6% character error rate for a wide range of string based CAPTCHA. If you can get a better score with this method, let me know!