**Python CAPTCHA Library Hacking Manual**

The code layout within the *tools* directory shouldn’t need modification, any further modification of the directories code structure may result in any errors arising.

1. **Environment**

* Operating System(s): **Windows 10, Ubuntu 16.04**
* Video Card(s): **Nvidia GTX 1070 8GB**
* RAM: **16GB**
* Anaconda Virtual Environment for Python codebase execution

1. **Dependencies**
   1. Python Version 3.6.8
   2. ‘requirements.txt’ satisfies package dependencies.
   3. Optional: We highly recommend that CUDA is installed along with its native libraries. ***This improved performance well over 800%!***
   4. Optional: Having FireFox installed allows the user to run ***live\_benchmark.py*** using the geckodriver.exe found within the ***utils/*** directory.
   5. On every new session, run the respective run file depending on your OS environment.
      1. Windows: run.bat
      2. UNIX: source run.src
2. **Generating & Training**
   1. Enter into the ***tools/generate\_train/*** directory.
   2. Execute ***python train\_fast.py -g NUMBEROFGENERATIONS -b BATCHSIZE -f FONTSDIRECTORY***
   3. The newly created/modified folder ***models*** will contain two files: ***cnn\_model.j5*** & ***cnn\_model.json***. Do not modify the names of these files, any benchmarking done with the two benchmarking tools rely on these specified names.
      1. This process trains and will reload/train a previous model ran through this process or the pre-existing model.
3. **Benchmarking**

To benchmark the newly created model, the ***benchmarking*** directory contains two scripts.

1. ***trained\_benchmark.py*** allows a pretrained model to be ran as a segmentation tool. Run this tool by having your model within the model’s directory.

**Warning!** Having a CUDA installed greatly improves the performance of this benchmarking tool.

* 1. This script utilizes the pretrained TensorFlow Object Detection (TOD) model that was pretrained. A setup for this process isn’t included but is referenced down in the ***Additional Notes*** section of this manual.

1. ***local\_benchmark.py*** runs the local *stored\_captchas/* directory and runs it against the captchas.txt in the root of the benchmarking directory. This is a way to run outside CAPTCHAs for testing purposes.
2. **live\_benchmark.py** runs on the web browser using a Selenium hook to obtain the imagery and element data.

**Warning!** This scripts website hook switched its CAPTCHA security to Google reCAPTCHA which broke our test session. Any other website using basic CAPTCHA may be properly implemented without knowing too much selenium.

**Additional Notes**

***utils/*** - Directory contains several helper scripts to ensure the data is sanitized.

* ***helper.py*** – Contains several functions for image processing and character extraction. This is the main framework for our developing segmentation procedure.
* ***mass\_verify.py*** – **Deprecated:** Script allowed for evaluation of character recognitions through a large quantity given the model and sheet.csv.
* ***reload.py*** – Allows for segmented portions through a .csv and the images directory present to be visualized, used to sanitize and find bad segments.
* ***verify\_cnn*** – Script allows any image passed in, of single character, to test the models recognition.
* ***Geckodriver***.exe – Hooks selenium to the FireFox webbrowser.