# Using Machine Learning to Break Visual Human Interaction Proofs (HIPs)

Kumar Chellapilla
kumarc@microsoft.com
Microsoft Research
One Microsoft Way
Redmond, WA 98052

Patrice Y. Simard
patrice@microsoft.com
Microsoft Research
One Microsoft Way
Redmond, WA 98052

## 1 Introduction

Human Interactive Proofs (HIPs) or Completely Automated Public Turing Tests to Tell Computers and Humans Apart (CAPTCHAs) are tests that enable the construction of automatic filters that can be used to prevent automated scripts from utilizing services intended for humans. These automated tests can be passed by humans but not by the current computers. Work on distinguishing computers from humans traces back to the original Turing Test which asks that a human distinguish between another human and a machine by asking questions of both.

## 2 Human Interactive Proofs (HIPs)

Most HIPs are pure recognition tasks which can be easily broken using machine learning. Harder HIPs are built using the combination of recognition and segmentation tasks.Hence segmentation is the most effective way to confuse the machine learning algorithms. Construction of HIPs that are of practical value is difficult because it is not sufficient to develop challenges at which humans are somewhat more successful than machines. This is because the cost of failure for an automatic attacker is minimal compared to the cost of failure for humans.
Specifically seven different HIPs namely Mailblocks, MSN, Ticketmaster, Yahoo, Yahoo v2, Register and Google are discussed.The methods for solving six of them are discussed.

## 3 Using machine learning to break HIPs

Breaking HIPs is not new; Mori and Malik have successfully broken the EZ-Gimpy (with 92% success) and Gimpy (with 32 % success) HIPs from CMU. Our approach aims at an automatic process for solving multiple HIPs with minimum human intervention, using machine learning. So, our main goal is to study the common strengths and weaknesses of these HIPs rather than to prove that we can break any one HIP in particular with the highest possible success rate. HIPs become harder when no language model is used. Similarly when a HIP uses a language model to generate challenges, success rate of attacks can be significantly improved by incorporating the language model. But language model is not common for all the HIPs, hence it is not discussed.

Our generic method for breaking all the HIPs is to write a custom algorithm to locate the characters and then use machine learning for recognition. Surprisingly, segmentation, or finding the characters, is simple for many HIPs which makes the process of breaking the HIP particularly easy. Once the segmentation problem is solved then it becomes a pure recognition problem and it can be trivially be solved using machine learning. It is stressed that our goal is not 100% success rate but something efficient that can achieve much better than 0.01%.

In each of the six HIPs discussed, 2500 HIPs were hand labeled and used as - (a) recognition [1600 for training, 200 for validation, and 200 for testing], and (b) segmentation [500 for testing segmentation]. For each of the five HIPs, a convolution neural network was trained and tested on gray level character images centered on teh guessed character positions. The trained neural network became the recognizer. Methods to solve each of the six HIPs is explained in detail along with their success rate.

## 4 Lessons learned from breaking HIPs

From the methods discussed it is clear that most of the errors come from incorrect segmentations, even though most of the development time is spent devising custom segmentaion schemes.

### 4.1 The segmentation problem

- Segmentation is computationally expensive.
- A Segmentation function is complex.
- For successful segmentation, our system must learn to identify which patterns are valid among the set of all the possible valid and non-valid patterns.

### 4.2 Building better/harder HIPs

Harder HIPs can be built by making the segmentation difficult. The idea is that the additional arcs are themselves good candidates for false characters.

The previous segmentation attacks would fail on this HIP. Despite the apparent difficulty of these HIPs, humans are far better than computers at segmentation.

### 4.3 Building an automatic segmentor

For building an automatic segmentor the following procedure is used: Label the characters based on their correct position and train the recognizer at all the locations in the HIP image. Collect all the candidate characters identified with high confidence by the recognizer. Compute the probability of each combination of candidates (from left to right), and output the solution string with highest probability. This method suggests another axis for comparing classifiers.

## 5 Conclusion

We learned that Segmentation plays a major role in building better/harder HIPs. Decomposing the process into recognition and segmentation simplifies the analysis. Recognition on even unprocessed images can be done automatically using neural networks. They have used this observation to design new HIPs and new tests for machine learning algorithms with the hope of improving them.
In this paper, we have successfully applied machine learning to the problem of solving HIPs.