Hacking Google reCAPTCHA v3 using Reinforcement Learning

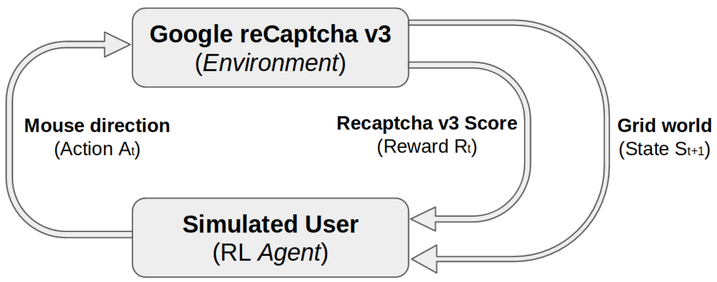
Ismail Akrout*, Amal Feriani*, Mohamed Akrout

TELECOM ParisTech

UNIVERSITY OF TORONTO

Summary

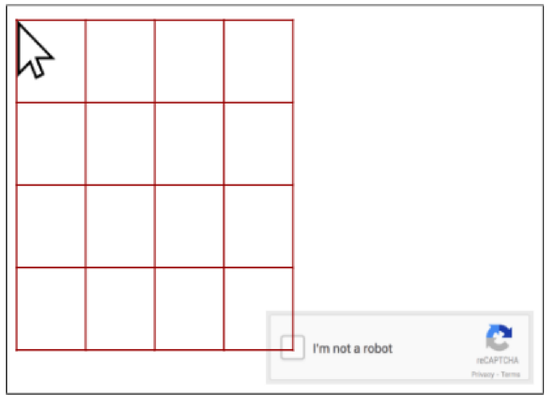
- Google's reCAPTCHA system, for detecting bots from humans, is the most used defense mechanism in websites. The newest version reCAPTCHA v3 uses ML techniques to return a risk assessment score to characterize the trustability of the user.
- **Research Question:** Is the ML-based version of reCAPTCHA vulnerable to automated attacks?
- **Idea**:
  - Use an RL agent to bypass Google reCAPTCHA v3



Problem Formulation

- The reCAPTCHA test is formulated as a MDP :
  - State space : The possible mouse positions in a web page
  - Action space : {left, right, bottom, up}
- train a RL agent to find optimal mouse trajectory from a random starting point to the reCaptcha checkbox.

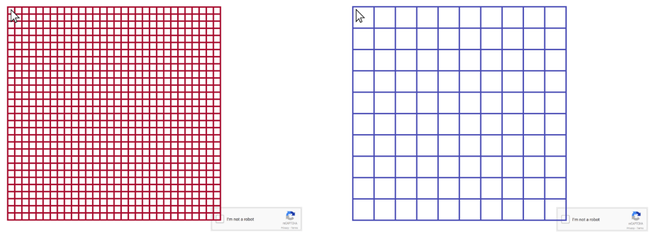
⇒ The problem is similar to a grid world problem



reCAPTCHA v3 Environment

- **Browser**:
  - does not use browser automation tools e.g. *Selenium*
  - is not connected to Google account
  - not connected using a proxy or VPN
- **Mouse**:
  - controlled using a dedicated Python package e.g. *PyAutoGUIlibrary*
- **Captcha v3 API**:
  - detects a system simulated with a browser automation tool even if a human user is using the mouse.
  - Detects IP rotation using API services such as *Tor*

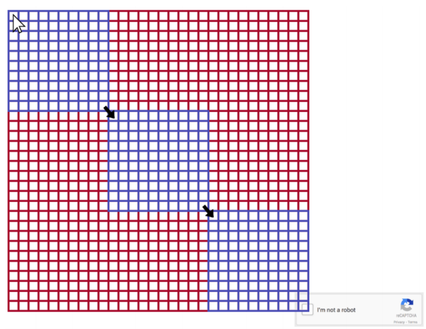
The grid world environment for different screen resolutions



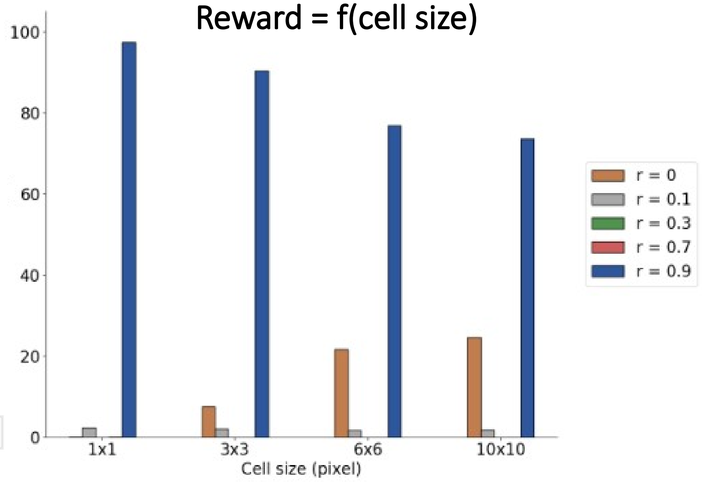
Results

- Only an environment design without detected automated tools provides high rewards.
- Train a **Reinforce** agent on a grid world environment with a specific size
- Use the obtained agent to choose optimal actions in the reCAPTCHA environment.
- Design an agent that successfully defeated the reCAPTCHA by obtaining a score of **0.9** for different cell sizes.

Illustration of the divide and conquer approach



Reward = f(cell size)



Reward = f(grid resolution)