COSE474-2020F: Final Project Proposal
"I'm a robot: Breaking Google reCAPTCHA using deep learning"

김건우 김기찬 민지수 정치훈

Abstract

This paper is the proposal of our final project; we aim to break through the Google reCAPTCHA without human interaction by training YOLO network model.

1. Introduction

The reCAPTCHA service offered by Google, is the most widely used captcha service, and has been adopted by many websites for preventing abusing by automated bots.

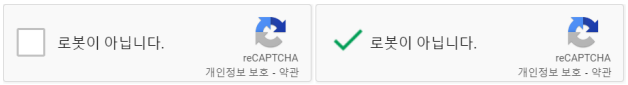

Figure 1. reCAPTCHAv2 widget. The widget will be checked if user is considered as human, otherwise it performs a similar images challenge.

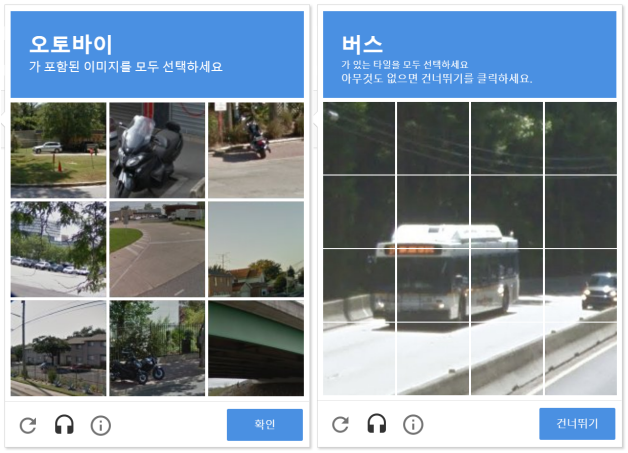

Figure 2. Similar images challenge by reCAPTCHA. There are 3 types; two of which are choosing from 9 independent images and an another is choosing part of 16-segmented image.

The current version, reCAPTCHAv2, is designed to be human-friendly and secure. Once the user clicks in the checkbox, a request is sent to Google containing the client's info such as cookie and browser info. The request is then analyzed by the system which decides what type of captcha challenge will be presented to the user. The different type of challenges are offered for each request.

We want to automate passing the reCaptcha, using web crawling and deep learning to break bot protection. Because of its similarity with object detection and image segmentation, we're trying to solve the challenge by applying YOLO v4 model.

2. Datasets

The images of reCaptcha challenges are road views, which consist of bus, crosswalk, traffic light, etc. We have a reCaptcha dataset uploaded by who researched first which contains about 10,000 images in each category, but unfortunately its amount and resolution is insufficient for training every types of the challenges. Thus, we decided to use additional dataset such as BDD100K, OpenImages V4 with modifying so that we may train our learning model with it.

3. Goals to achieve throughout this project

We'll produce object detection model that recognize keywords come up with the challenge, and then implement automated bot via Selenium solving the reCaptcha verification process using trained model.

4. Schedule and Role

Model investigation with testing SOTA code performance (10/5∼12) : Gichan Kim, Jisu Min

Image dataset standardization for training (10/13∼25) : Geonu Kim, Chihoon Jung

Prior research investigation for implementing model (10/13∼25) : Gichan Kim, Jisu Min

Implementing object detection model (10/26∼11/8) : Gichan Kim, Geonu Kim, Jisu Min, Chihoon Jung

Testing and optimizing our model (11/9∼29) : Gichan Kim, Geonu Kim, Jisu Min, Chihoon Jung

Implementing automation process (11/9∼15) : Gichan Kim, Geonu Kim, Jisu Min, Chihoon Jung

5. Comparison with SOTA

References

[1] L. Von Ahn, B. Maurer, C. McMillen, D. Abraham, and M. Blum, reCAPTCHA: Human-based character recognition via web security measures, *Science*, vol. 321, no. 5895, 2008.

[2] A. Bochkovskiy, C. Wang, H. M. Liao, YOLOv4: Optimal Speed and Accuracy of Object Detection, arXiv:2004.10934

[3] F. Yu, W. Xian, Y. Chen, F. Liu, M. Liao, V. Madhavan, T. Darrell, BDD100K: A Diverse Driving Video Database with Scalable Annotation Tooling, arXiv:1805.04687

[4] A. Kuznetsova, H. Rom, N. Alldrin, J. Uijlings, I. Krasin, J. Pont-Tuset, S. Kamali, S. Popov, M. Malloci, A. Kolesnikov, T. Duerig, V. Ferrari, The Open Images Dataset V4: Unified image classification, object detection, and visual relationship detection at scale, arXiv:1811.00982