KTH ROYAL INSTITUTE
OF TECHNOLOGY

DEGREE PROJECT IN COMPUTER SCIENCE AND ENGINEERING

FIRST CYCLE, 15 CREDITS

# Image Recognition for Solving Google's reCAPTCHA

## An Investigation of how Different Aspects Affects the Security of Google's reCAPTCHA

**JENNIFER LARSSON**
**OSCAR BERGSTRÖM**

Stockholm, Sweden 2022

# Image Recognition for Solving Google's reCAPTCHA

JENNIFER LARSSON
OSCAR BERGSTRÖM

Bachelor in Computer Science
Date: June 7, 2022
Supervisor: Jörg Conradt
Examiner: Pawel Herman
School of Electrical Engineering and Computer Science
Swedish title: Bildigenkänning för att lösa Google's reCAPTCHA

# Abstract

In this project, an image recognition solver for Google's reCAPTCHA was used to investigate how different aspects of the reCAPTCHA challenge affects its efficiency on defending against bots. The aspects that were examined was the type of reCAPTCHA challenge and the object which should be identified in the images. To conduct the study, an image recognition algorithm, trained on a large image dataset, was used in a script that interacted with a reCAPTCHA website. Testing showed that both the investigated aspects had a large impact on the efficiency, individually and in combination.

# Sammanfattning

I detta projekt användes en bildigenkännings-algoritm för Google's reCAPTCHA för att undersöka hur olika aspekter av en reCAPTCHA-utmaning påverkar dess effektivitet i att skydda hemsidor mot botar. De aspekter som testades var typen av reCAPTCHA utmaning samt objektet som skulle identifieras i bilderna. För att genomföra studien användes en bildigenkännings-algoritm som var tränad på ett stort dataset med bilder i en script som interagerade med en reCAPTCHA hemsida. Testning visade att bägge de undersökta aspekterna hade en stor påverkan på effektiviteten, både individuellt och i kombination.

# Contents

# Chapter 1

# Introduction

Internet security is a crucial subject, and its importance grows as the world becomes more connected. Tests to tell humans and computers apart has become a necessity on the internet. One approach is the CAPTCHA (Completely Automated Turing test to tell Computers and Humans Apart), which is a security mechanism used to protect websites from bots. The main idea is that in order to use the website, the user has to pass a test, which should be easy to solve for humans and difficult for computers. The first CAPTCHAs were text-based, where warped text was presented to the user who should identify what it said. However, as computers evolved and became better at reading warped text, the text-based CAPTCHA wasn't efficient enough in defending against bots. Consequently, Google's reCAPTCHA v1 was shut down in 2018 [1]. Google released version two of their reCAPTCHA in 2014, an image based CAPTCHA where the user should identify objects in images. In the beginning, this type of CAPTCHA was more effective, but is now being challenged by the recent years development in machine learning algorithms [2]. As computers become better at image recognition, it is interesting to investigate how secure the image based CAPTCHA really is, and what aspects affect the safety.

## 1.1 Purpose

The purpose of this report is to investigate which images are more effective in defending against bots in Google's reCAPTCHA v2. The project aims to identify which attributes of an image that makes it difficult for a bot to identify. Apart from the attributes that make an object difficult to identify, this project also aims to identify which type of Google reCAPTCHA challenge is more difficult to solve and what aspects of it contribute to why. It is an interest-

ing subject to research because Google's reCAPTCHA v2 is one of the most common defense mechanisms used on the internet, and it is facing a growing challenge with machine learning algorithms that can identify objects in images.

Previous research has been done on the subject. One study, which solves three different types of CAPTCHA with machine learning, was conducted in 2016. The solver in this study had a sucess rate of 70,8 % [3]. Another study, which uses an object detection based solver for solving Google's reCAPTCHA v2, was conducted in 2020 and had a sucess rate of 83,3 % [4]. This project's contribution to the subject is comparing the efficiency of different types of images and challenge types, and researching why some are more secure than others. This is a valuable contribution because it can provide deeper understanding about what issues are affecting Google's reCAPTCHA v2 and how these issues can be addressed.

## 1.2 Research Question

This study aims to investigate the following:

- How is the efficiency of an image recognition solver for Google's reCAPTCHA influenced by the object which should be identified in the images and the type of reCAPTCHA challenge?

## 1.3 Scope

The type of captcha will be limited to Google reCAPTCHA version 2. Version 2 contains multiple different ways to test whether or not a user is human. This study will only include the visual "I'm not a robot"-checkbox test. Therefore human-like cookie handling and interactions will not be necessary for this study.

The study is limited to only test objects that are present in the Common Objects in Context dataset. Objects that are not present in this dataset will not be taken into account, since a reload of the reCAPTCHA will result in a new challenge. Therefore it is not necessary to implement the full list of challenges from reCAPTCHA.

The project is also limited by the type of image recognition algorithm. A detection based algorithm called YOLO (You Only Look Once) will be used in this study.

## 1.4 Outline

The background chapter starts with a description of how image recognition works and detailed explanations of the YOLO algorithm and the COCO dataset. Further on in the background, Google's reCAPTCHA is described and its different challenge types. In the methodology chapter, the implementation and the associated tools of the study are described. Further on in the report, the results chapter displays the findings of this study, together with visual representations. These results and their effects are discussed later in the report, in the discussion chapter. In the final chapter, the conclusions of this study are summarized.

# Chapter 2

# Background

## 2.1 Image Recognition

Image recognition is a technology used to enable computers to identify objects in digital images and videos. This technology is a sub-category of computer vision, which is a broader term that includes several methods to gather, process, and identify real-world data. Apart from image recognition, computer vision also includes for example video tracking and image reconstruction [5].

### 2.1.1 Categories of Image Recognition

Image recognition can be divided into several categories, seen in figure 2.1, which are used for different tasks. The categories are:

- **Classification**: Identifies the class of an image. An image can only belong to one class.
- **Tagging**: Identifies multiple objects in an image. An image can have multiple tags.
- **Localization**: Identifies the class of an image, localizes the object in the image and creates a bounding box around it.
- **Detection**: Identifies multiple objects in an image, localizes the objects and creates bounding boxes around each identified object.
- **Semantic Segmentation**: Identifies the class of an image and localizes the object to the nearest pixel.

- **Instance Segmentation**: Identifies multiple objects in an image and can differentiate multiple objects belonging to the same class [5].


Figure 2.1: Different types of image recognition [6]

### 2.1.2 Deep Neural Networks

A digital image can be represented with a matrix of numbers, where each element represents the intensity at each pixel. This data and the corresponding labels are fed to the image recognition system, which can be trained to identify patterns and relations between the data from different images. After the training is completed, the system's accuracy can be analyzed on test data [5].

Image recognition is implemented using Deep Neural Networks (DNNs), which lets the system learn by example. Neural Networks consist of three types of layers - input, hidden, and output layers. The layers can be seen in figure 2.2. The input layer receives the input, the hidden layers process it, and the output sends out an output signal which includes a decision about the input data. A layer is made up of artificial neurons, which are interconnected nodes that perform the calculations. Traditional Neural Networks usually have up to three hidden layers, in contrast to DNNs which can have hundreds [6].

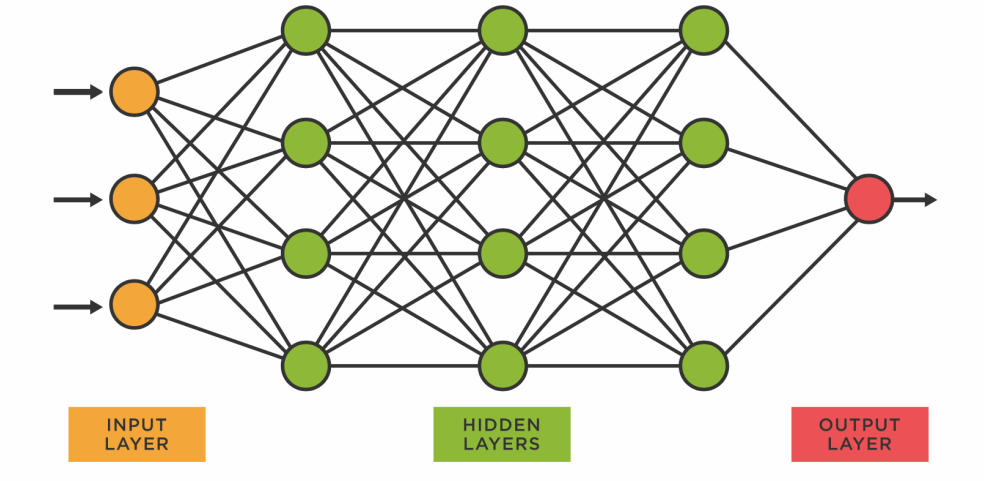


Figure 2.2: The layers of a neural network [7]

### 2.1.3 Convolutional Neural Networks

Convolutional Neural Networks (CNNs) are a type of DNNs used primarily for image recognition. The CNN architecture consists of three different types of hidden layers - convolutional, pooling and fully connected layers. The neurons in the convolutional layers calculate a scalar product, resulting in a decreased vector. The pooling layers reduce the size of the vector by performing down-sampling along the height and width of the input. The fully-connected layers produce a one-dimensional vector where each element is a class score. The output of the system is the class predictions [8]. The architecture of a CNN is outlined in figure 2.3.

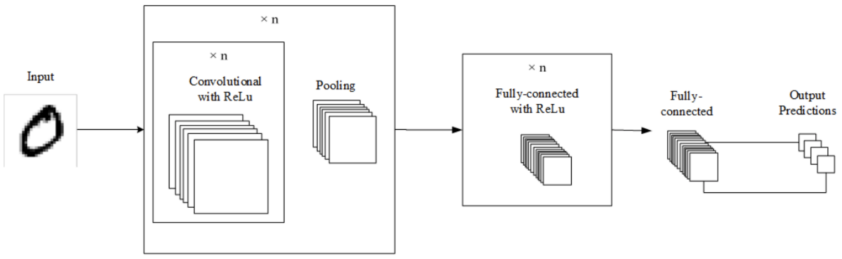


Figure 2.3: CNN architecture [8]

## 2.2 YOLO

YOLO stands for You Only Look Once, and is a detection-based image recognition architecture. The first version was released in 2015, and since then four following versions have been released. YOLO reframes image recognition to a single regression problem, and goes directly from image pixels to bounding boxes and class probabilities. The system only looks at the image once to identify which objects are present in the image and their location. One single CNN simultaneously predicts multiple bounding boxes and class probabilities for each box [9].

Compared to traditional object detection methods, YOLO has several benefits. Because YOLO treats detection as a regression problem, there is no need for an advanced pipeline. This makes YOLO very fast and can detect objects in real time, while achieving higher accuracies than other real time systems. Secondly, YOLO uses the entire image when making class predictions, which gives less background errors than methods that view the image in different regions. Furthermore, YOLO is very generalizable and is therefore not likely to crash when faced with unexpected input [9].

### 2.2.1 Implementation of YOLO

The YOLO algorithm makes use of three techniques to detect objects in images - Residual Blocks, Bounding Box Regression, and Intersection Over Union (IOU).

Firstly, the image is divided into a grid of residual blocks, see figure 2.4, where each grid cell has the same dimensions. The purpose of the grid is that each residual block is responsible for detecting the objects in them [10].


Figure 2.4: An image divided into grids [10]

Each grid cell predicts the objects within them and a bounding box is produced around each object, see figure 2.5. Single bounding box regression is used to predict the attributes of a bounding box. The bounding box has the attributes height, width, center, and class [10].

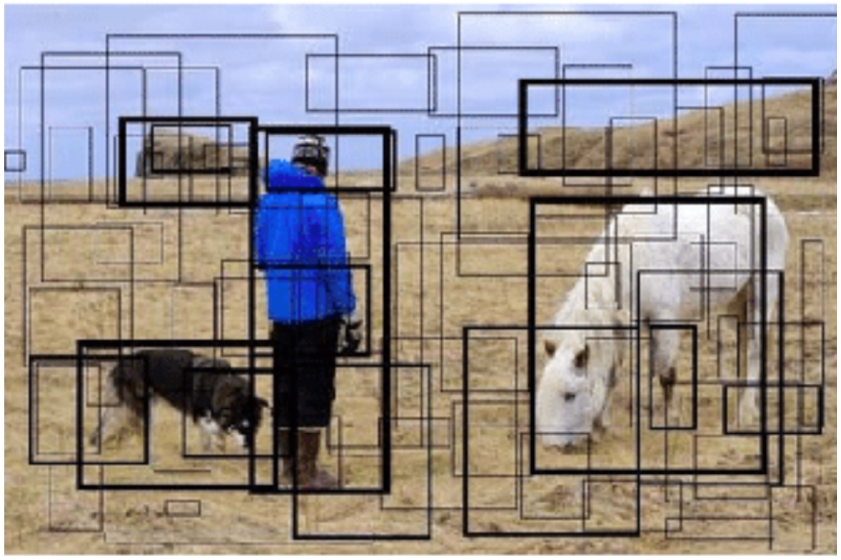

Figure 2.5: Bounding boxes [11]

After the bounding boxes have been created, YOLO makes the final class prediction using IOU, which is a tool used to eliminate bounding boxes that are not equal to the real boxes. This results in removing the bounding boxes which do not match the actual objects [10]. The stages of YOLOs detection progress can be seen in figure 2.6.

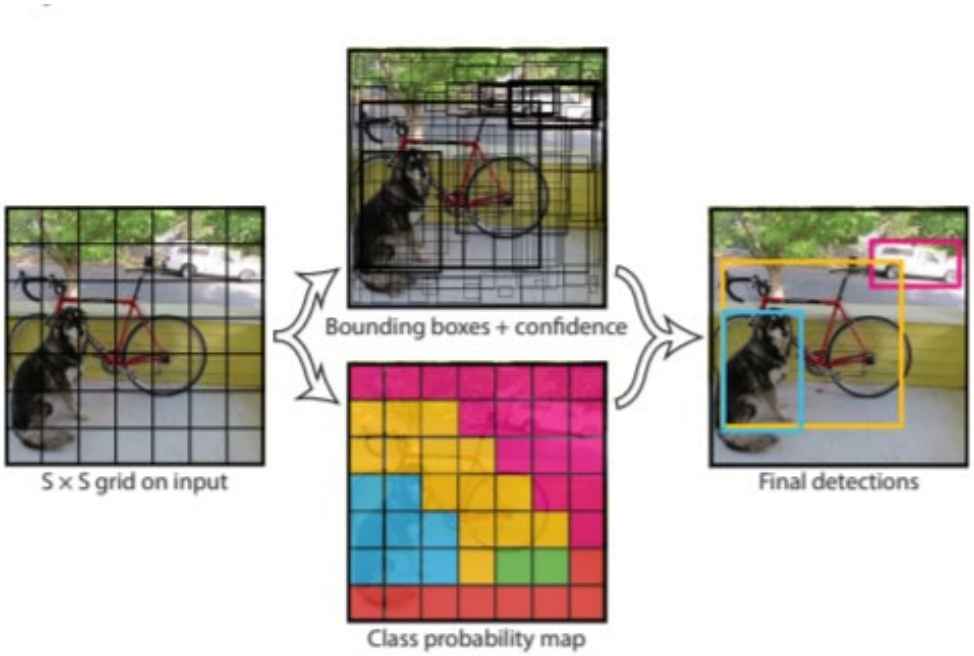


Figure 2.6: The different stages of the YOLO object detection algorithm [10]

### 2.2.2 Versions of YOLO

The first version of YOLO was released in 2015 and was the first object detection network of its kind. The second version was released two years later and featured some improvements, for example higher resolution and anchor boxes [12]. A year later, the third version was released. YOLO v3 added an objectiveness score and made predictions on three layers [13]. The fourth version, released in 2020, included improvements in for example feature aggregation [14]. The fifth and most recent version, also released in 2020, utilizes Pytorch training procedures [15].

## 2.3 Common Objects in Context Dataset

Common Objects in Context (COCO) is a large-scale object detection, segmentation, and captioning dataset. The COCO dataset is made with the goal of advancing object recognition. This is done by placing the question of object recognition in the context of the broader question of scene understanding [16].

Other datasets made for computer vision mainly use one of the following segmentations: image classification, object localization, or semantic segmentation. But COCO uses Instance Segmentation seen in (d) in figure 7. It does this in order to create a dataset composed of images depicting complex everyday scenes of common objects in their natural context [16]. It also marks the area and position of the object which enables the dataset to be used where object localization segmentation, (b) in figure 2.7, is used.

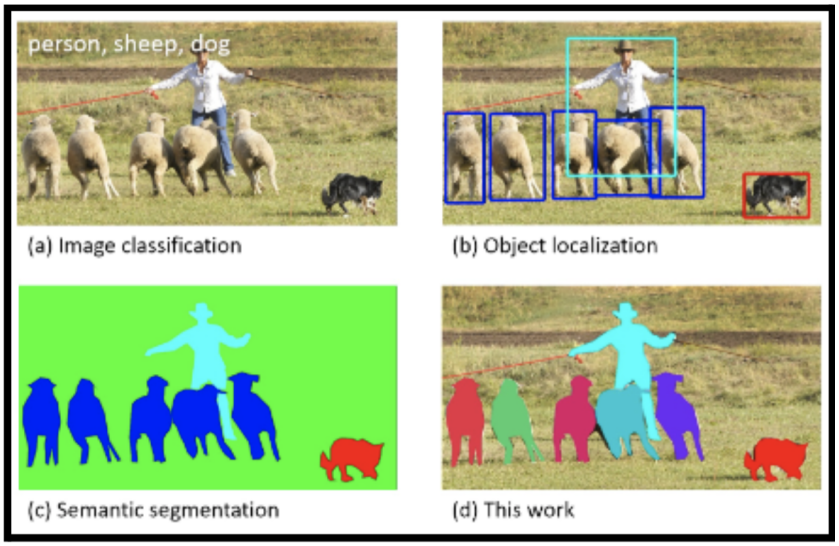


Figure 2.7: Different types of segmentation used in datasets for image recognition training [16]

COCO consists of 164 000 labeled images. Within these images there are 1.5 million labeled objects. COCO is a relatively small dataset in terms of classes, there are 80 different classes. Examples of possible classes are ‘car’ , ‘tennis racket’ or ‘pizza’ [17]. These predefined classes are useful if it is not necessary for the algorithm to train on all objects, since it is easy to get a group of classes from the dataset. This can make training the algorithm faster.

The images are divided into three sets specifically made for training, validating and testing the algorithm, which streamlines the process of training a

machine learning algorithm. The training dataset contains 118 000 images, the validation set is 5 000 images and the testing set is 41 000 images [17].

## 2.4 Google's reCAPTCHA v2

reCAPTCHA is a service developed by Google to hinder bot and spam attacks. It does this by providing different tasks for the user to complete. The tasks are developed in a way that should be easy to solve by humans and difficult for computers.

Early versions of reCAPTCHA had tests where the computer distorted words or random letters so that a bot would not be able to pattern match the letters on the image to characters. This was enough protection for a while, but machine learning algorithms got better at reading distorted words which prompted Google to develop a new version of reCAPTCHA that would have greater protection against the improved bots. So in 2014 Google released reCAPTCHA v2. It challenged the user to find and pick objects in pictures.

### 2.4.1 Challenge Types

There are three different but similar challenges in reCAPTCHA v2. All consisting of a grid where the user should select the boxes according to the challenge prompt.

#### 4x4 Grid Challenge

The 4x4 grid is one picture divided into tiles inside a grid, see figure 2.8. The user is prompted to choose all of the tiles containing a certain object. When all of the tiles are chosen the user will then press the blue button which will either approve the user or prompt another challenge.

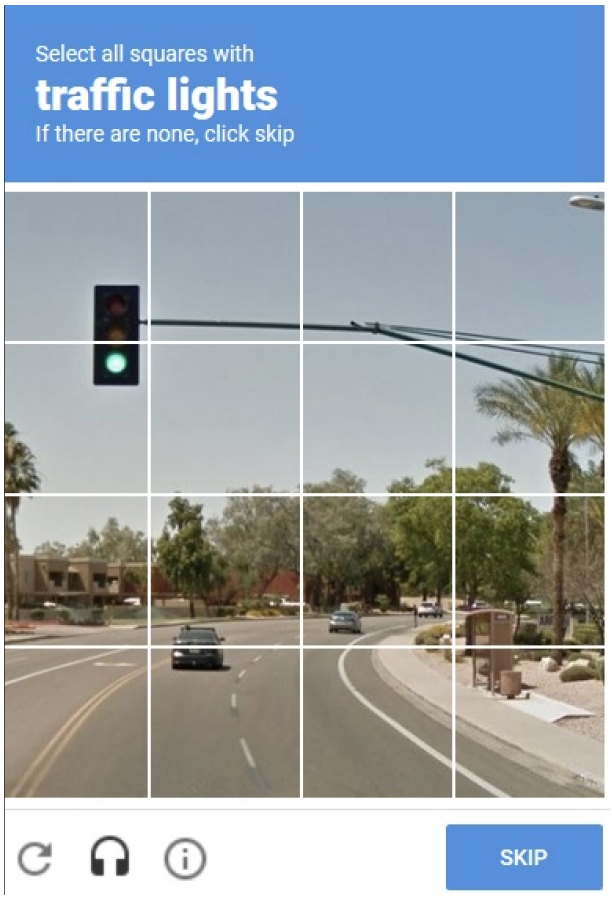


Figure 2.8: 4x4 grid challenge

### 3x3 Grid Without Fading Challenge

The 3x3 without fading challenge contains nine different pictures, see figure 2.9. The user is prompted to choose all pictures containing a certain object. When all of the pictures are chosen the user will press the blue button which will either approve the user or prompt another challenge.

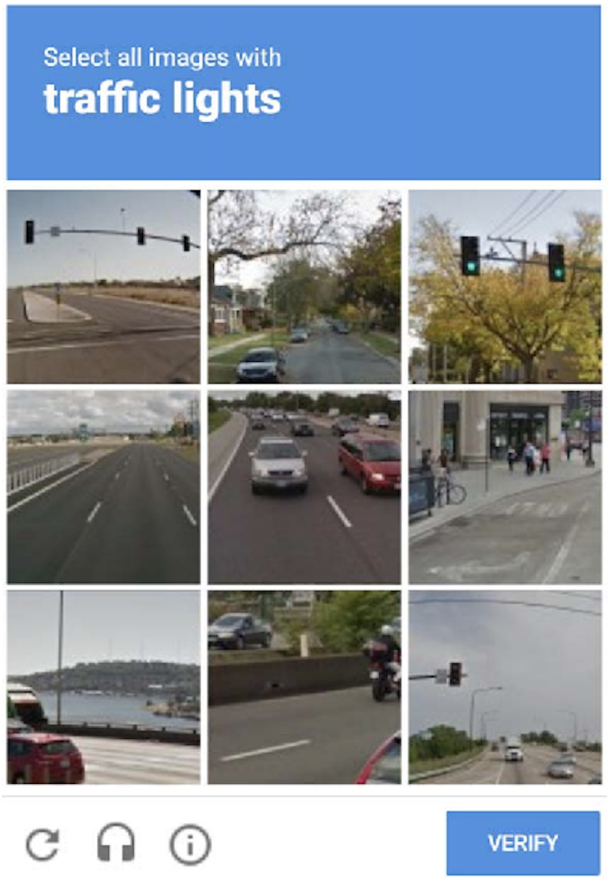


Figure 2.9: 3x3 grid without fading challenge

### 3x3 Grid With Fading Challenge

The 3x3 with fading challenge contains nice different pictures, see figure 2.10. The user is prompted to choose all pictures containing a certain object. When a picture is chosen it will fade out and be replaced with another picture. The user needs to continue the challenge until no images containing the object are left. When the user is done they must press the blue button which either approves the user or prompts another challenge.


Figure 2.10: 3x3 grid with fading challenge

### 2.4.2 reCAPTCHA Development

These challenges were too advanced for common machine learning algorithms during the time, so reCAPTCHA v2 was able to protect websites against bot and spam attacks where version 1 failed.

Later throughout the development of reCAPTCHA v2, tests were added to check whether or not the cookies of the user passed as human. This was implemented in order to improve the user experience. If the cookie tests fail then the user will have to pass a regular image challenge instead.

# Chapter 3

# Methods

## 3.1 Manual Testing

In order to get an overview of how common different objects and challenge types were, 150 manual tests were performed in the initial phase of the development. The data from this testing was used to adapt the script, which is described in section 3.5.

## 3.2 reCAPTCHA on Localhost

To get access to Google's reCAPTCHA v2, a simple website was created with PHP and JavaScript. The website was hosted locally and only contained the reCAPTCHA. This website was used by the script described in section 3.5 to gather data in order to answer the research question of this report.

## 3.3 Selenium Webdriver

Selenium is a library used for automating web applications, which includes the Selenium Webdriver. The Webdriver is a collection of open source APIs which are used for automating testing of web applications. The Webdriver can be used to find elements on a webpage by different attributes, such as ID, class name and XPATH. These attributes can be found for all elements on a page using the inspect element tool. In this project, Selenium Webdriver was used in the script described in section 3.5 to open websites, find elements on the page and click on them.

## 3.4 YOLO and COCO

To detect the objects in the reCAPTCHA images, YOLO version 5 was used with pretrained weights on the COCO dataset. YOLOv5 has multiple different image recognition models which are suitable for different datasets, and in this project YOLOv5l (version 5 large) was used. The large model is suitable for larger datasets, such as COCO, and has a higher accuracy, but it is slightly slower than the simpler models of YOLOv5.

## 3.5 Script

### 3.5.1 Setup

In order to connect reCAPTCHA with the image recognition algorithm, a python script was written. First, the script opened a new incognito Google Chrome tab with Selenium Webdriver. The tab had to be in incognito mode to avoid cookies which could have an impact on the reCAPTCHA challenge. Then, the script used the Webdriver to click on the checkbox, identify the challenge type, and get the object which should be identified. The checkbox can be seen in figure 3.1, and the challenge type and object in figure 3.2.

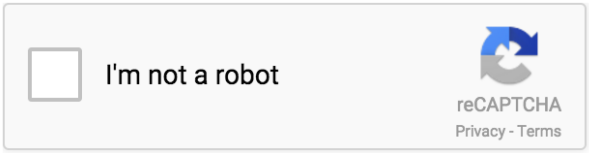


Figure 3.1: reCAPTCHA checkbox

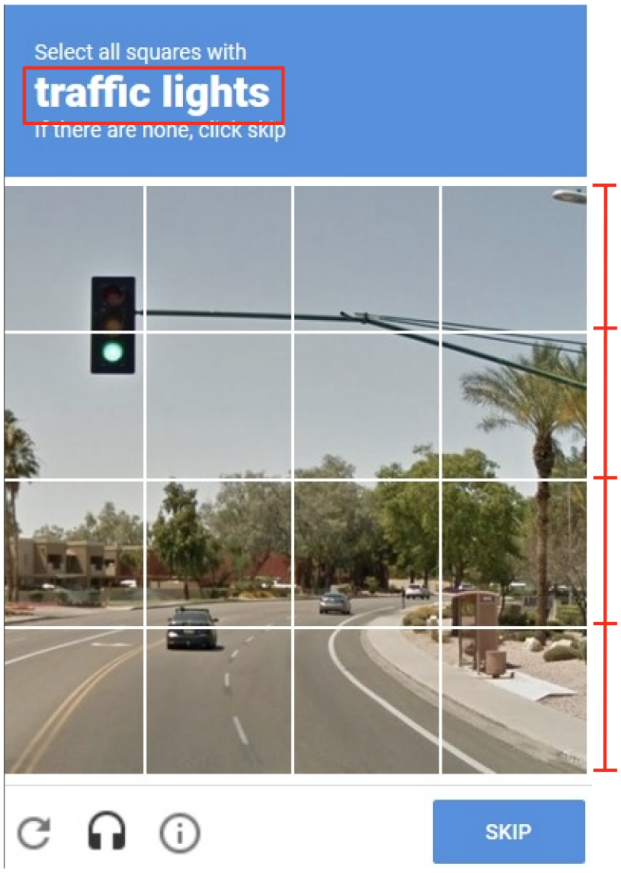


Figure 3.2: reCAPTCHA object and challenge type

### 3.5.2 Challenge Types

Three different functions were implemented for solving the different challenge types. These functions are described in this section.

#### 4x4 Grid Challenge

For the 4x4 challenge type, the image was downloaded and used in the YOLO algorithm to detect which object the image contained. If the object matched the object which should be detected, the position, height and width of the bounding box surrounding the object was used to find which tiles the object included. These tiles were clicked and then the submit button was clicked to get the next image and the same procedure was repeated until the challenge was solved.

### 3x3 Grid Without Fading Challenge

To solve the 3x3 grid without fading challenge, the image was downloaded and the objects in the image were detected with YOLO. The positions of the objects which matched the object which should be identified were found and the corresponding tiles were clicked. Then the submit button was clicked.

### 3x3 Grid With Fading Challenge

In order to solve the 3x3 grid with fading challenge, the image was downloaded and used with YOLO. The positions were found and corresponding tiles were clicked. After the clicked tiles had faded out and new images replaced them, these images were downloaded and used with YOLO individually. If any of them matched the object which should be identified, they were clicked. This procedure was repeated until there were no more matching images, and the submit button was clicked.

## 3.6 Testing

Because the research question involves how the efficiency of Google's reCAPTCHA is affected by the object which should be identified and the challenge type, the results from each individual image challenge were collected. This is because solving one entire reCAPTCHA challenge can involve solving multiple images, with different objects and challenge types. To get a result with the data relevant for the research question, the challenge type, object and success of 250 image challenges were gathered.

The data was used to calculate the success rates for different objects and challenge types, as well as the overall success rate. These results were analyzed to draw conclusions about how the efficiency of an image recognition solver for Google's reCAPTCHA is influenced by the object which should be identified in the images and the type of reCAPTCHA challenge.

# Chapter 4

# Results

## 4.1 General Findings

After testing 250 challenges, 118 challenges were successful and 132 were failed. Therefore, the average success rate was 47,2%, which can be seen in figure 4.1.

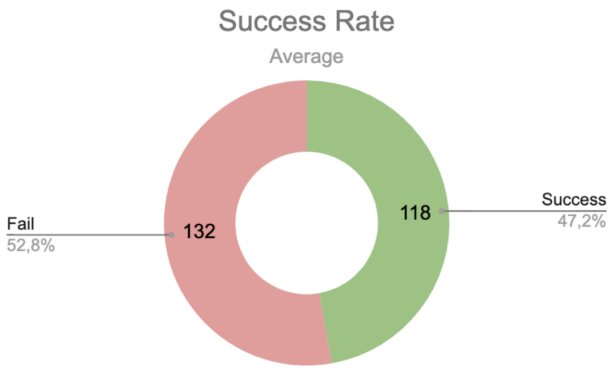


Figure 4.1: Average success rate over all tests

## 4.2 Challenge Types

There was a significant difference in the accuracy of the different challenge types and how often the different challenge types were displayed. In this section, the results from the different challenge types will be presented.

### 4.2.1 4x4 Grid Challenge

Out of 250 tests, 107 were the 4x4 grid challenge. 84 of these were successful, and 23 were failed. This gave a success rate of 78,4%, which can be seen in figure 4.2.

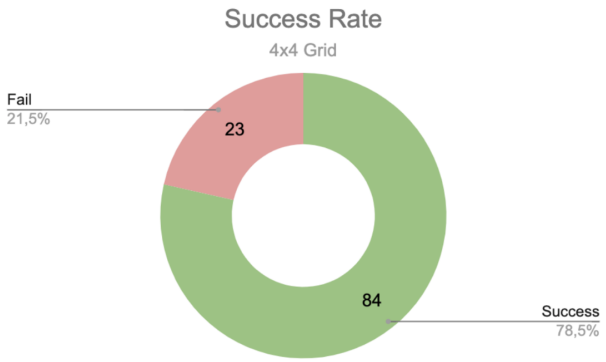


Figure 4.2: Success rate for the 4x4 grid challenge

### 4.2.2 3x3 Grid Without Fading Challenge

None of the tests were of the challenge type with a 3x3 grid without fading. Therefore, this challenge type will not be discussed further in the report.

### 4.2.3 3x3 Grid With Fading Challenge

143 of the 250 tests were the 3x3 with fading challenge, and out of these 34 tests were successful and 109 failed. The success rate was 23,8%, which is shown in figure 4.3.

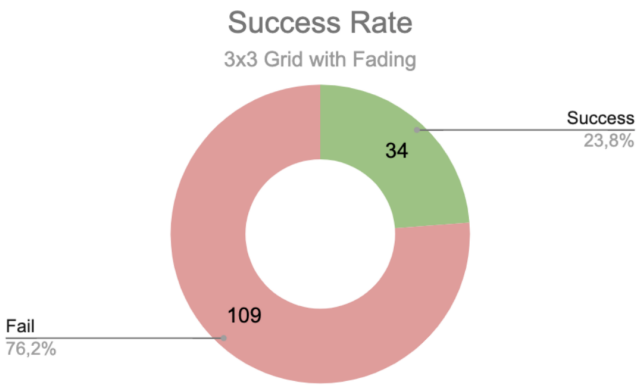


Figure 4.3: Success rate for the 3x3 with fading challenge

## 4.3 Objects

### 4.3.1 Non-present Objects

The frequency of objects in the testing were contrasting to the result in the manual testing. So some objects such as buses were more frequent in the testing than in the manual tests and other objects were more rare. Due to this some objects that were present during the manual testing were not present during the actual testing. The relevant objects, which YOLO would be able to detect, were parking meters and boats.

### 4.3.2 General Findings

The success rate for different objects for both challenges varies. From the results in figure 4.4 it is possible to disclose the following. Traffic lights had the highest success rate at 81% and cars had the lowest success rate at 3%. Traffic lights had the highest frequency with 69 attempts and motorbikes had the lowest frequency at 11 attempts.

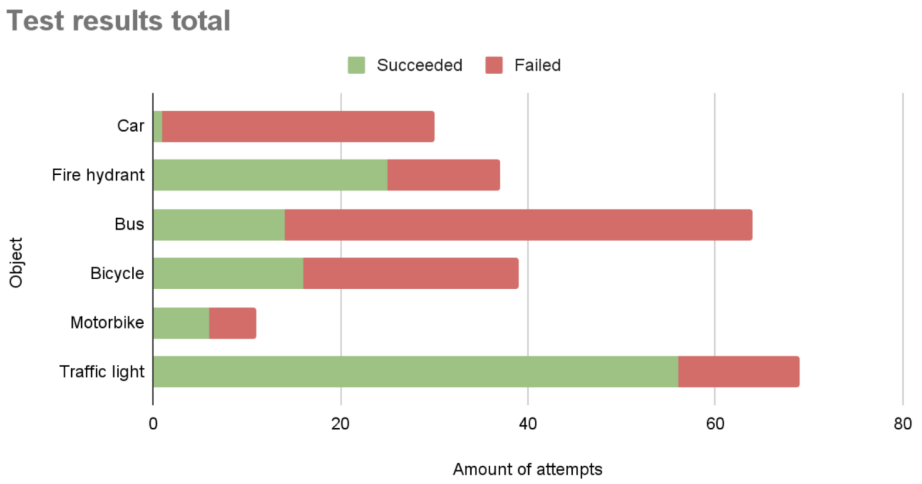


Figure 4.4: Average success rates for different objects

### 4.3.3 4x4 Grid Challenge

The success rate for different objects for the 4x4 challenge were relatively uniform. From figure 4.5, one can deduct that traffic lights had the highest success rate at 81% and motorbikes had the lowest success rate at 55%. Traffic lights were the most frequent object at 69 attempts and cars were the least frequent object at zero attempts.

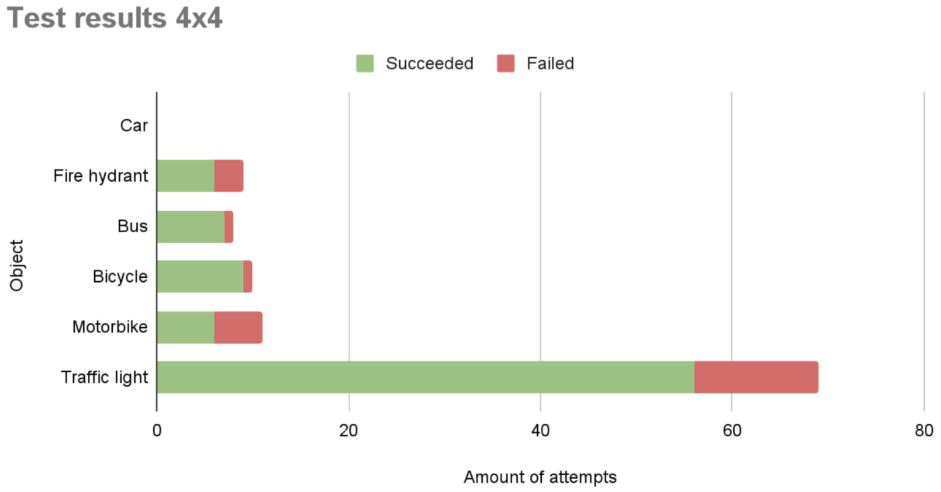


Figure 4.5: Success rates for different objects in the 4x4 grid challenge

### 4.3.4 3x3 Grid With Fading Challenge

The success rate for different objects for the 3x3 challenge has an outlier. From figure 4.6, one can deduct that fire hydrants had the highest success rate at 68%. Cars had the lowest success rate of 3%. Buses had the highest frequency at 56 attempts. Traffic lights and motorbikes had the lowest frequency at zero attempts.

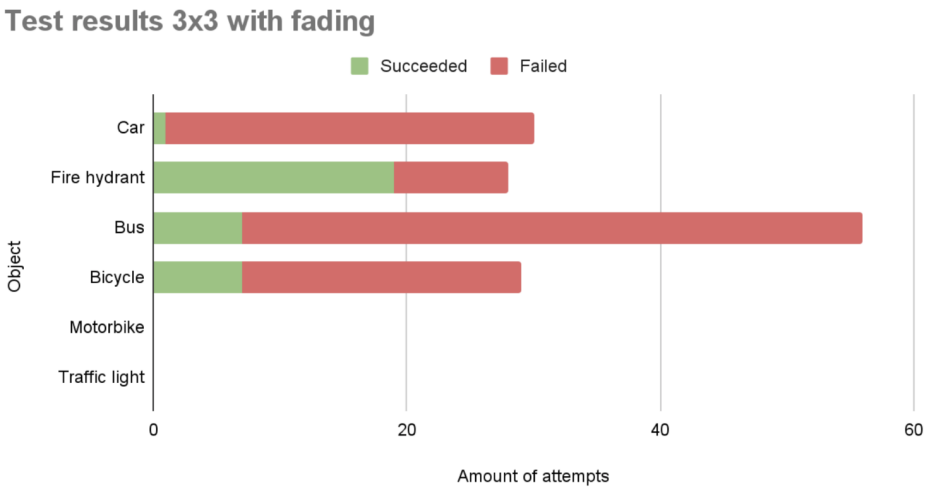


Figure 4.6: Success rates for different objects in the 3x3 grid with fading challenge

# Chapter 5

# Discussion

The average accuracy of the tests was 47,2%, which means that the YOLOv5 solver trained on the COCO dataset could successfully solve around half of the challenges. However, there were significant differences in the accuracies for different challenge types and objects which should be identified in the images. In this section of the report, these results and the possible explanations for them will be discussed.

## 5.1 Challenge Types

There was a big difference in how well the solver could solve different challenge types. The success rate of the 4x4 grid challenge was 78,5%, which is much higher than the success rate for the 3x3 grid with fading challenge, which was 23,8%. There are many possible reasons for this difference. However, two main aspects that differentiate the challenge types have been identified and will be discussed in section 5.1.1-2.

### 5.1.1 Frequency of Objects

A 4x4 grid challenge typically consists of one object covering multiple grid cells or two to three smaller objects covering one grid cell each. The grid cells which contain the object which should be identified or a part of that object should be clicked in order to solve the challenge. This means that the solver in most cases only has to identify one or a couple of objects. This differs from the 3x3 grid with fading challenge, where the solver has to often have to identify more objects in order to succeed. There are nine grid cells and usually around half of them contain the object. After these tiles have been clicked, new

images, which can also contain the object, replace them. Consequently, the solver often has to identify more objects in the 3x3 with fading challenge than the 4x4 challenge in order to succeed, which could have a significant effect on the accuracy of the different challenge types.

### 5.1.2 Image Size and Quality

Another aspect that differs between the challenge types are the image size and quality. The 4x4 grid challenge displays a single image which can be used in the YOLO detect algorithm. In contrast to the 3x3 grid with fading challenge, which displays nine smaller images, which together are the same size as the single image in the 4x4 grid challenge. The solver has to detect if the object is present in each of the smaller images and the challenge is failed if the solver fails to detect all images correctly. This means that the 3x3 grid challenge requires more of YOLO and is more difficult than the 4x4 grid challenge. Consequently, the accuracy of the 3x3 grid challenge is lower.

Apart from the image size, the image quality also differs between the challenge types. The images in the 3x3 grid are typically more grainy and have a lower resolution, in contrast to the image in the 4x4 grid, which are higher quality. This makes the 3x3 grid challenge more difficult, which yields a lower success rate.

## 5.2 Objects

The results show that the success rate varied between different objects. There were also some differences between certain objects in the 3x3 grid challenge with fading and 4x4 grid challenge. The main explanation for the difference in success rate between objects is the typical attributes an object possesses in the reCAPTCHA images.

Some objects depend more on their surroundings to be identified which is difficult for YOLO to take into account. In contrast to humans, who look at the whole perspective rather than specific objects of the image, it is possible for humans to identify the object and pass the challenge.

For some objects the issue was shape. If the object did not have a rectangular shape it could cause the bot to choose too many tiles in the 4x4 grid challenge. The reason for this is that YOLO uses bounding boxes, so the object was always identified as a rectangle which caused empty tiles to be chosen by the bot.

The results for each object and the possible explanations for them are discussed in section 5.2.1-6.

### 5.2.1 Cars

Cars had the lowest success rate out of any object at 3,3%. There are mainly two probable factors for this. First of all cars only appeared in the 3x3 grid challenge which had a significantly lower success rate than the 4x4 grid challenge. Second of all is that humans can easily distinguish cars from the surroundings. As seen in figure 5.1 the car is difficult to identify when taken out of context. But when the whole image is revealed it is easy to identify that the object should be a car. This is because humans can gather information and identify that there is an object present on the road, it is smaller than a bus or a truck and larger than a bicycle. Therefore it is likely to be a car. This thought process is difficult for YOLO to do. This means that cars are effective objects to use to distinguish bots from humans.

Figure 5.1: The same car out of context is more difficult to identify

### 5.2.2 Buses

Buses had the second lowest success rate at 21,9%. The difference of results for the 3x3 grid challenge and the 4x4 grid challenge were significant. Buses had a success rate of 87,5% for the 4x4 grid challenge and 12,5% for the 3x3 grid challenge. Buses were significantly more frequent in the 3x3 grid challenge. Consequently, the difficulty in identifying buses was not the only reason for the low overall success rate. The results were dependent on how frequent buses were in the different challenge types. The overall results for buses were therefore not completely representative.

The results for buses in the 3x3 grid challenge were the second lowest. This is likely because of the same reason as for cars. Humans can use the surroundings to get the context of the image which simplifies the task of identifying the object. This is difficult for YOLO to do, especially in the smaller images

in the 3x3 grid challenge. Therefore buses are an effective object to use in reCAPTCHA's 3x3 grid challenge, but less effective in the 4x4 grid challenge.

### 5.2.3 Fire hydrants

Fire hydrants had the second highest success rate and the highest success rate out of any object in the 3x3 grid challenge. Fire hydrants performed marginally better in the 3x3 grid challenge than the 4x4 grid challenge and was the only object to do so. This is because of the shape of a fire hydrant. The typical round top of a fire hydrant led to excess tiles being chosen due to the bounding box including area around the fire hydrant as seen in figure 5.2. This affected the 4x4 grid challenge results negatively but had no effect on the 3x3 grid challenge results. All failed 4x4 tests were due to excess tiles being included in the bounding box.

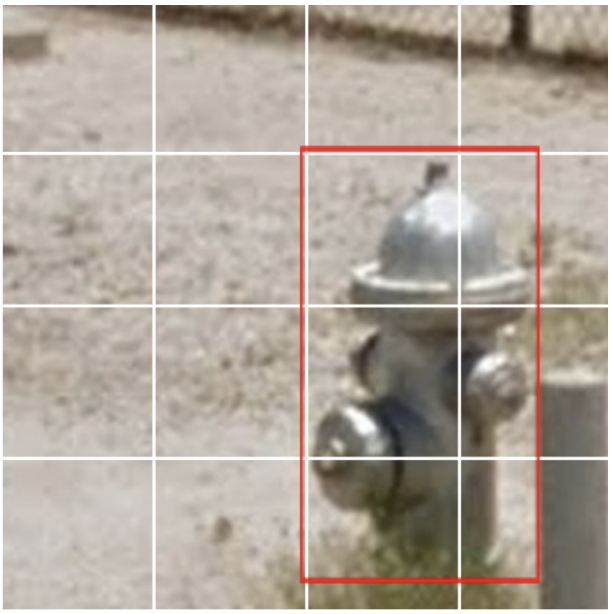

Figure 5.2: A fire hydrant with a bounding box covering too many grid cells

Fire hydrants are often very distinct colors, such as red, silver and yellow, which often have a large contrast to the background. This could help the bot to identify the object and may be a reason for the high success rate in the tests. Overall fire hydrants are ineffective objects to use in the reCAPTCHA.

### 5.2.4 Bicycles

Bicycles were present in both challenges and had an overall success rate of 41,0%. In the 3x3 grid challenge, the success rate was 24,1% and in the 4x4 grid challenge, the success rate was 90,0%. One attribute of the bicycle which could have had a negative impact on the success rate is that bicycles often had another object blocking some of the bicycle, which makes it more difficult for YOLO to identify the bicycle correctly. In fact 71,4% of the failed bicycle challenges had some object or person blocking a part of the bicycle. This was more frequent in the 3x3 grid images. Bicycles also have slimmer features than the other objects, specifically the tires and the bicycle frame. Consequently, these features are often difficult to distinguish even for humans. Overall bicycles are an effective object in the 3x3 grid challenge and an ineffective object for larger images of the 4x4 grid challenge.

### 5.2.5 Motorcycles

Motorcycles were only present in the 4x4 grid challenge. Motorcycles had an overall success rate of 54,5% and had the lowest success rate for objects in the 4x4 grid challenge. One possible reason for the relatively low success rate could be due the shape of a motorcycle. As seen in figure 5.3, there are multiple areas inside the bounding box where the motorcycle is not present. This causes the script to choose tiles which should not be chosen. Out of the failed attempts 80,0% of them were due to excess tiles being chosen as a result of the bounding box including areas where no motorcycle were present. This makes motorcycles an effective object to use in the 4x4 grid challenge.

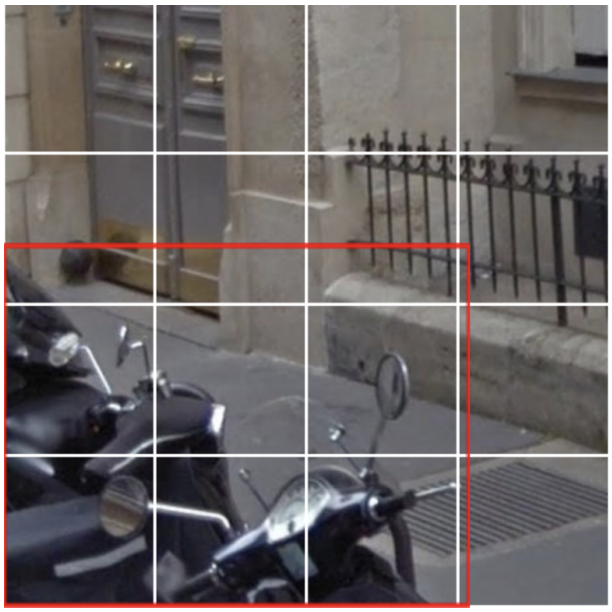

Figure 5.3: A motorcycle with a bounding box covering too many grid cells

### 5.2.6 Traffic Lights

Traffic lights were only present in the 4x4 grid challenge, and had the highest success rate of any object at 81,2%. In comparison to other objects in the 4x4 grid challenge, traffic lights had the third highest success rate, after buses and bicycles respectively.

One attribute that may hinder YOLO to successfully detect traffic lights in the challenges is that images with traffic lights often contain multiple traffic lights as seen in figure 5.4. This means that the bot is tested on identifying multiple objects and if it fails to identify a single traffic light it would fail the entire challenge. This is unique for traffic lights, where other objects typically only occur once in the 4x4 grid challenge. The bot managed to identify at least one traffic light in 61,5% of the failed attempts.

Figure 5.4: A 4x4 grid challenge with multiple traffic lights

One attribute of traffic lights which makes them easier to identify by YOLO is that they are typically rectangular. Consequently, the bounding box often matches the shape very well, as seen in figure 5.5.

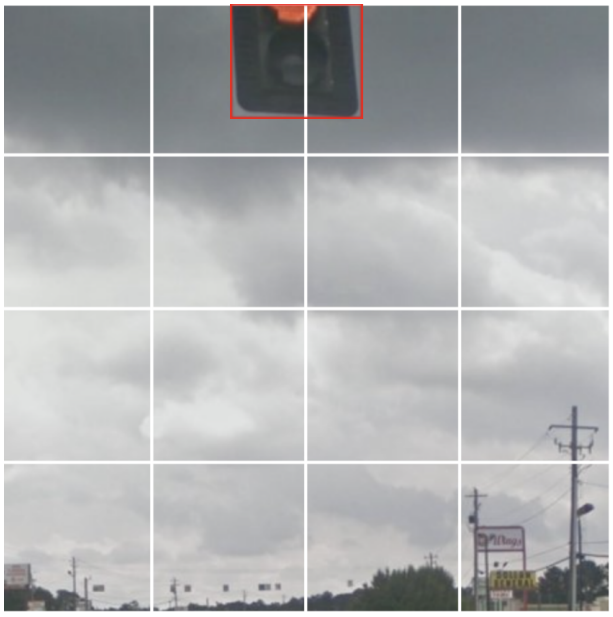

Figure 5.5: A traffic light which matches the bounding box well

## 5.3 Comparison to Previous Research

As mentioned in the introduction, previous research has been done on the subject. The image recognition solvers used in the previous studies had success rates of 70,8 % and 83,3 % respectively, which differs from this studies success rate of 47,2 %. There are multiple possible explanations for this, which will be discussed in this section.

The first is the choice of image recognition algorithm. As stated earlier, YOLO creates a bounding box around the object. If the objects shape differs a lot from the bounding box, excess tiles will be clicked and the challenge will not be successfully solved. None of the previous studies used YOLOv5, which partially can explain the difference in success rates.

The second is the frequency of different challenge types. The solver in this study had much higher success rate for the 4x4 grid challenge. However, only 42,8 % of the challenges were of this type. The rest was the 3x3 grid with fading challenge, which was significantly more difficult for the solver. This challenge type distribution made the overall success rate much lower than the 4x4 grid success rate.

However, the purpose of this study was not to create the most successful solver for Google's reCAPTCHA. The aim was to investigate how different aspects of the reCAPTCHA affects its efficiency in defending against bots. Therefor, the most interesting aspect of the results is not the overall success rate, but rather how the success rate differ between different challenge types and objects.

## 5.4 Implications

Google's reCAPTCHA is one of the most common internet security tools that exists today, but as the machine learning field develops and image recognition algorithms become more efficient, internet security has to develop as well. In this section, the possible implications of this project and how they correspond with the development of reCAPTCHAs will be discussed.

The results from this study indicates that some reCAPTCHA challenges are more difficult for a detection based solver to solve. For example, the testing showed that the 3x3 grid challenge was more difficult than the 4x4, and that cars were more difficult than traffic lights. These indications can provide useful information about how different aspects of a reCAPTCHA challenge can be altered to make reCAPTCHAs more secure. Therefore, the challenges which

were found to be easily solved could be altered or replaced to provide better internet security.

## 5.5 Ethics

The purpose of a reCAPTCHA is to give the user a test that should be easy to solve for humans and difficult for computers. It is not only important to defend against bots, but also to not disturb the user experience. This is a difficult tradeoff where the importance of user experience has to be weighed against the security of the website. For some websites, for example websites with a lot of private or valuable information, the security aspect is more important. For other, for example websites with an entertainment purpose, the focus is on the user experience. The way that Google's reCAPTCHA v2 is structured, improving one aspect can have a negative impact on the other. Therefore, it is a valid question whether interactive reCAPTCHAs are facing their downfall and should be replaced by a reCAPTCHA that does not affect the user experience. This trend can already be seen, as Google's reCAPTCHA v2 is steadily being replaced by the third version, which gives the user a score based on how it uses the website and does not require any user interaction.

## 5.6 Further Research

### 5.6.1 Image Recognition Algorithm

The results in this study were affected by the choice of image recognition algorithm. As previously stated, YOLO detects objects in images and produces a bounding box around its edges with the attributes width, height and center coordinates. If an object does not have a rectangular shape, the bounding box will include more than only the object. In the 4x4 grid challenge, a bounding box which covers more than the object could cover more grid cells than it should. Some examples of objects which were affected by this are motorcycles and bicycles, see section 5.2.

To avoid this problem in further research and create more effective image recognition bots for solving Google's reCAPTCHA v2, another type of image recognition algorithm could be used. One approach is instance segmentation, which is explained in section 2.1.1, which localizes the object by the nearest pixel and does therefore not include unnecessary space around the object. Consequently, the grid cells clicked by the script in the 4x4 grid challenge

would be less likely to not include the object to be identified.

### 5.6.2 Comparison Between Humans and Computers

This project focuses on what attributes make the reCAPTCHA more effective in defending against bots. However, the study has only gathered data about how efficient the detection based solver is for different objects and challenge types, without comparing these results to human results. If an object is very difficult for computers to identify, one could assume that it is an effective object in defending against bots. However, the same object could also be as difficult for humans to identify, which defeats the purpose of the reCAPTCHA. The challenge should be difficult for computers but easy for humans, and if it is difficult for both it is not an effective challenge. To gain more perspective and to better interpret the results in future research, one could give humans and computers the same tests and compare the results.

# Chapter 6

# Conclusions

From the results in this study, the conclusion can be drawn that the challenge type and the object which should be identified in the images have a large impact on how efficient Google's reCAPTCHA v2 is in defending against a detection based solver. The 4x4 grid challenge was less effective than the 3x3 grid challenge with fading. Generally, cars were much more efficient than traffic lights, and fire hydrants, buses, bicycles and motorcycles were relatively equal. However, there was a big difference in the efficiency of some objects for different challenge types. Overall, the efficiency of a challenge was affected by the combination of the challenge type and the object which should be identified in the images.

# Bibliography


[1] Google. *Choosing the type of reCAPTCHA*. URL: https://developers.google.com/recaptcha/docs/versions#v1. (accessed: 21.04.2022).

[2] Kaiming He et al. "Surpassing humanlevel performance on imagenet classification". In: (2015).

[3] Suphannee Sivakorn, Iasonas Polakis, and Angelos D Keromytis. "I Am Robot: (Deep) Learning to Break Semantic Image CAPTCHAs". In: (2016).

[4] Imran Hossen et al. "An Object Detection based Solver for Google's Image reCAPTCHA v2". In: (2020).

[5] Maruti Techlabs. *What is the Working of Image Recognition and How is it Used?* URL: https://marutitech.com/working-image-recognition/. (accessed: 21.04.2022).

[6] Altexsoft. *Image Recognition with Deep Neural Networks and its Use Cases*. URL: https://www.altexsoft.com/blog/image-recognition-neural-networks-use-cases/. (accessed: 21.04.2022).

[7] TIBCO. *What is a Neural Network?* URL: https://www.tibco.com/reference-center/what-is-a-neural-network. (accessed: 21.04.2022).

[8] Keiron O'Shea and Ryan Nash. "An Introduction to Convolutional Neural Networks". In: (2015).

[9] Joseph Redmon et al. "You Only Look Once: Unified, Real-Time Object Detection". In: (2016).

[10] Grace Karami. *Introduction to YOLO Algorithm for Object Detection*. URL: https://www.section.io/engineering-education/introduction-to-yolo-algorithm-for-object-detection/. (accessed: 22.04.2022).

[11] Abidha Pandey, Manish Puri, and Aparna S Varde. "Object Detection with Neural Models, Deep Learning and Common Sense to Aid Smart Mobility". In: (2018).

[12] Joseph Redmon and Ali Farhadi. "YOLO9000: Better, Faster, Stronger". In: (2016).

[13] Joseph Redmon and Ali Farhadi. "YOLOv3: An Incremental Improvement". In: (2018).

[14] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. "YOLOv4: Optimal Speed and Accuracy of Object Detection". In: (2020).

[15] Jacob Solawetz. *YOLOv5 New Version - Improvements And Evaluation*. URL: https://blog.roboflow.com/yolov5-improvements-and-evaluation/. (accessed: 22.04.2022).

[16] Tsung-Yi Lin et al. "Microsoft COCO: Common Objects in Context". In: (2015).

[17] COCO. *Common Objects in Context*. URL: https://cocodataset.org/#home. (accessed: 24.04.2022).

TRITA-EECS-EX-2022:494