Can Today's Machine Learning Pass Image-Based Turing Tests?*

Apostolis Zarras[1], Ilias Gerostathopoulos[2], and Daniel Méndez Fernández[2]

[1] Maastricht University, Maastricht, The Netherlands
[2] Technical University of Munich, Munich, Germany

Abstract. Artificial Intelligence (AI) in general and Machine Learning (ML) in particular, have received much attention in recent years also thanks to current advancements in computational infrastructures. One prominent example application of ML is given by image recognition services that allow to recognize characteristics in images and classify them accordingly. One question that arises, also in light of current debates that are fueled with emotions rather than evidence, is to which extent such ML services can already pass image-based Turing Tests. In other words, can ML services imitate human (cognitive and creative) tasks to an extent that their behavior remains indistinguishable from human behavior? If so, what does this mean from a security perspective? In this paper, we evaluate a number of publicly available ML services for the degree to which they can be used to pass image-based Turing Tests. We do so by applying selected ML services to 10,500 randomly collected CAPTCHAs including approximately 100,000 images. We further investigate the degree to which CAPTCHA solving can become an automated procedure. Our results strengthen our confidence in that today's available and ready-to-use ML services can indeed be used to pass image-based Turing Tests, rising new questions on the security of systems that rely on this image-based technology as a security measure.

1 Introduction

Artificial Intelligence (AI) has been coined by pioneers like Alan Turing in the 1950's [35] and deals ever since with the fundamental effort "*to automate intellectual tasks normally performed by humans*" [6]. One core area of AI is Machine Learning (ML) where—in contrast to rather classical instruction-based programming in which machines process given datasets based on predefined rules—machines are *trained* with large datasets to recognize representation patterns in the data and produce the processing rules, thus, they "learn" how to recognize and classify given phenomena [6].

Thanks to recent advancements in computational infrastructures and the availability of large datasets that are fundamental to ML, artificial intelligence has been making long and decisive strides forward from the 1990's on. These advancements are made along two main paths: (*i*) the research in introducing new

* Final authors' version

and improving existing ML techniques and methods (e.g., deep learning, convolutional neural networks, Gaussian processes) and (*ii*) the widespread adoption of ML techniques and methods in both research and practice. As for the latter, there exist nowadays many "ML-as-a-service" offerings, which simplify the access to and the use of powerful ML-enabled functionalities.

A representative example of one such type of offering is given by image recognition services. A number of providers, from large companies such as Amazon, IBM, Google, and Microsoft, to startups such as Clarify and Cloudsight, offer paid services allowing other companies or individuals to add advanced image recognition capabilities to their systems. Such capabilities include, inter alia, classifying/labeling an arbitrary image with a number of tags at certain confidence levels, determining whether an image contains a given element (object/person), or finding similar images in a collection.

Fueled by, at least from an application perspective, major advancements in machine learning, we can witness very optimistic marketing slogans accompanying available services ("build apps that see the world like you do" [7]). Needless to say, also negative future scenarios on threats potentially imposed by ML are heavily spread in the public sphere [29]. In fact, today's public debates are too often comparable to a hype full of emotions and conventional wisdom rather than rational debates on basis of concrete evidence on the state of the practice and reasonable implications this has on security issues. Without any prejudice and expectations on future applications of AI, one interesting and important question yet remains: How far we have actually come as of today with current technologies? In other words, could current ML advancements pass the Turing Test, i.e., could they imitate human (cognitive and creative) tasks to an extent that their behavior remains indistinguishable from human behavior?

To the best of our knowledge, there exists little evidence about the extent to which Machine Learning currently can pass Turing Tests and the implications this has on topics like security. Indeed, there has been so far no systematic attempt to validate and compare the effectiveness and applicability of ML techniques in controlled settings.

With this paper, we contribute a curiosity-driven study with the aim to provide a first step in closing the knowledge gap on the state of ML with respect to (image-based) Turing Tests. In essence, our goal is to critically evaluate a number of ML services by the degree to which they can be used to pass Turing Tests. This shall allow to critically reflect upon the security implications that current advancements in AI and in ML have.

Turing Tests are embodied in the latest versions of widely used CAPTCHA services (e.g., Google's reCAPTCHA). Image-based CAPTCHAs rely on the assumption that a specific task, in this case that of image recognition, is presumingly difficult for AI but easy for humans based on their cognitive abilities and experiences. If the capabilities of currently available cloud-based ML services suffice to solve such problems, creating an automatic solver for image-based CAPTCHAs by relying on these services would be technically feasible and even economically viable. A consequent question therefore is for us: To which extent do image-based

CAPTCHAs still pose a reliable Turing Test and what are the security implications?

The reason behind choosing image-based CAPTCHAs as our benchmark is manifold. First, they provide a neutral ground for comparing the different image recognition services, as none of these services is tailored to breaking CAPTCHAs, i.e., to pass the Turing Test based on image recognition. A further reason is of pragmatic nature: CAPTCHAs are, same as ML services, largely available to the public facilitating studies, replications, and the public discourse. Finally, we consider it important that the demonstration facilitates a discussion on a larger scale since it shall put forward important security considerations for the future of ML in general, but also of CAPTCHAs in particular. We consider a re-evaluation of mechanisms such as ones incorporated in the de-facto standard CAPTCHAs to be important, because of their criticality to the security of many of today's systems.

In summary, we make the following main contributions:

- We investigate the effectiveness of in total *six* image recognition ML services.
- We design a system capable of accurately solving CAPTCHAs by leveraging the aforementioned services.
- We discuss the impact and implications on the security of systems relying on CAPTCHAs.

2 Fundamentals

In the following, we discuss the fundamentals to the extent necessary in context of our study. More precisely, we first provide information regarding the advances in ML and how these can be used for image recognition. Next, we briefly introduce how these image recognition algorithms are embodied in cloud-based services. Finally, we provide details of the current state of CAPTCHAs.

2.1 Image Recognition via Machine Learning

The ML technology empowering almost all of the image recognition tasks is deep learning, i.e., learning using information processing architectures with several layers [12]. One particular architecture, widely-known as convolutional neural networks (CNNs) [26], has proven very effective in image classification and object detection [12]. Its application relies on the existence of large amounts of annotated image data, from which a classifier is trained by iteratively learning higher-level features from lower-level ones. CNNs consist of multiple layers of convolution and pooling. While convolutional layer extracts features from data samples by moving the convolution filter in a predefined window, pooling layer takes the results of a convolutional layer as input and extracts the most important features. The convolutional filter is used in recognizing distinct objects in an image, almost invariant of their position. Research in image recognition with CNNs is fueled by the ImageNet annual competition [31]. Apart from image

recognition, deep learning has been successfully applied in other fields such as speech and audio recognition, natural language processing, machine translation, and even malware detection [10, 17, 23, 24].

Although image recognition via the latest machine learning techniques mentioned above can produce excellent results, it requires (*i*) considerable expertise in the ML algorithms, (*ii*) the availability of large datasets for training, and (*iii*) the operation of the necessary (typically GPU-enabled) infrastructures. The use of image recognition services lifts these assumptions.

2.2 ML Image Recognition Services

To provide a quick start in using image recognition for several business needs (e.g., social media photo tagging, digital asset management, or identification of common problems in health images), several image recognition as-a-service offerings have emerged. These are cloud-hosted services that require an image and provide one or more of the following functionalities: (*i*) Annotating the image with a set of labels, according to detected objects, living beings, scenes, and actions; (*ii*) Searching for similar images in a repository or in the Web; (*iii*) Categorizing the image according to a predefined taxonomy; (*iv*) Detecting and analyzing faces (including identifying age, gender, and/or emotional state) in the image; (*v*) Detecting celebrities, landmarks, logos, and/or inappropriate (violent, adult) content in the image; (*vi*) Detecting and extracting text in the image via Optical Character Recognition.

Since 2015, there has been a growth in the number and quality of publicly-accessible commercial image recognition services [16]. Such services are provided by large companies such as Amazon [1], IBM [19], Google [14], and Microsoft [28], but also smaller ones such as Clarifai [7], Cloudsight [9], Imagga [21], scale [33], Crimson Hexagon [20], Saltlab World [32], Jastec [22], and Cliq Orange [8].

Apart from using their pre-trained ML classifiers in providing the functionalities listed above, some providers allow the creation of custom classifiers or "models", upon provision of labeled datasets. This way, more specific business needs can be met, for instance, related to the analysis of a particular type of images. There are also companies that focus exclusively on such custom image recognition classifiers and APIs, most notably hive.ai [18] and Vize.ai [36].

2.3 CAPTCHA

The idea of discriminating humans from computers by letting them apply sensory and cognitive skills to solve simple problems, which have proven to be extremely hard for computer software, goes back to 1997 [30]. The term CAPTCHA (i.e., *Completely Automated Public Turing Test To Tell Computers and Humans Apart*) was first introduced by von Ahn et al. in an attempt to create automated tests that humans could pass and computer programs could not [37]. The main application of CAPTCHAs has been the detection of bots that perform malevolent activities such as generating large amounts of emails or accounts, participating in online polls, or posting messages in popular services.

There exist different types of CAPTCHA challenges, each requiring a human end-user to perform a specific cognitive task. The most common type requires the user to identify the characters of a distorted text box (text-based CAPTCHAs). Other common options include transcribing speech (audio-based CAPTCHAs) and identifying images that belong to a particular category (image-based CAPTCHAs). In any case, CAPTCHAs rely on a hard underlying AI problem, in particular, that of text, speech, or image recognition. As a result, apart from security reasons, CAPTCHAs are being used also as benchmarks for AI technologies.

There have been several attempts to create automatic solvers of CAPTCHAs from security researchers. So far, both text-based and audio-based CAPTCHAs have proven vulnerable to different attacks [3,4]. That is why popular and widely-used CAPTCHA implementations, such as Google's reCAPTCHA, are shifting towards image-based CAPTCHAs. In our work, we focus on solving image-based CAPTCHAs using publicly available ML image recognition services.

3 Study Design

Our overall objective is to better understand the extent to which image-based CAPTCHAs still pose a reliable Turing test and what are the implications on security. To this end, we formulate the set of research questions described below before introducing the data collection and analysis procedures.

3.1 Research Questions

To achieve our overall objective, we first need to understand what the potential of ML is with respect to image-based CAPTCHAs and accordingly design our study along three research questions:

***RQ1*:** What is the precision and recall of ML services?
***RQ2*:** What is the absolute accuracy of ML services when considering breaking CAPTCHAs?
***RQ3*:** What is the sufficient accuracy of ML services when considering breaking CAPTCHAs?

First, we want to understand what is the precision and recall of given ML services to recognize the images included in the CAPTCHAs (*RQ1*). This allows us to obtain a basic understanding on the general potential of the services. As the images strongly differ in the content they represent (e.g., a river versus a car), we want to further understand whether there are differences with respect to the particularities of the images themselves and what they represent respectively. Once we understand the general potential of the ML services, we want to analyze the extent to which they can be used to "break" CAPTCHAs, i.e., how well available services can be trained to bypass the today's widespread image-based Turing tests.

We do so in two steps (*RQ2* and *RQ3*): First, by analyzing the *absolute accuracy* of the services in terms of their potential to correctly classify all images of single CAPTCHAs into correct answers to that CAPTCHA or not. Second, image-based CAPTCHAs, if used in context of security mechanisms such as login mechanisms, usually allow for a specific failure tolerance (e.g., by allowing to classify one image wrongly). To lay the ground for our second contribution discussing the impact on security issues, we want to know what the *sufficient accuracy* of the services is in breaking CAPTCHAs. Given that this discussion is based, in parts, on analytical work, we also provide a brief discussion of the security impact analysis procedure to the extent necessary to reproduce our work (Section 5).

3.2 Data Collection and Analysis

In the following, we introduce the data collection and the analysis procedures used in this study.

Data Collection: The raw dataset of our study consists of the images contained in 10,500 image CAPTCHAs. For our study, we use six image-recognition services (i.e., *Google's Cloud Vision*, *IBM's Watson Visual Recognition*, *Amazon's Rekognition*, *Microsoft's Computer Vision*, *Clarify's Visual Search*, and *Cloudsight*). The selection of these services is made based on their popularity. To conduct our study where we compare these image-recognition services, we have to first prepare the data by establishing an oracle (i.e., ground truth) which we use to train a meta-classifier for each ML service-CAPTCHA category pair. Next, we employ the services and analyze the results with respect to our research questions.

In brief, to prepare the dataset for our study, we (*i*) retrieve the images contained in 10,500 CAPTCHAs (99,108 images), (*ii*) manually solve the CAPTCHAs in the sense of annotating `TRUE`/`FALSE` labels to the embedded images, and finally (*iii*) submit each image to each of the image recognition services, retrieve, and store the results. In the following, we provide further details for each of the aforementioned steps.

First, we leverage Google's reCAPTCHA service [13] to create our corpus of image CAPTCHAs. For ethical reasons and to not interfere with the traffic of a legitimate website, we set up a reCAPTCHA challenge on a website created for the sole purpose of this study. Next, we scrap the contents of the reCAPTCHA challenges to retrieve the embedded images; we repeat the process 10,500 times. To automate the process, we utilize *Selenium*, a software-testing framework for web applications that has the ability to programatically control a real web browser, in our case Google Chrome. During each challenge, we store in a MongoDB database all the information regarding the category of the challenge, (e.g., "Select all images with street numbers") and the individual images. Since reCAPTCHA returns a single image file and the image grid (e.g., "3x3"), we crop the larger image according to the grid to obtain the individual images. Such an exemplary cropped image can be seen in Figure 1.

Second, we go through the collected 10,500 challenges and manually solve them by marking the images that are correct answers to each challenge with a `TRUE` flag. It is worth mentioning here that not all the images have an unambiguous semantics (e.g., a building can have also a store). Sometimes is equally difficult for a human to solve a CAPTCHA as it is for an automated system. Hence, in cases where the actual semantics of an image are not completely clear, a majority vote determines the final solution.

Fig. 1. An exemplary image that was marked as correct answer to the challenge of category "Select all images with cars."

Third, we apply the image-recognition services to automatically label each of the collected images with metadata describing this image. As such, we leverage the image labeling APIs of the six different image recognition services we evaluate. Each API requires a jpeg-encoded file as input and provides a JSON-encoded response with metadata in the form of labels, concepts, classes, tags, or captions. For the remainder of the paper, we refer to these as *keywords*. With the exception of Cloudsight, all services provide a numeric value capturing the confidence value or score of each keyword, to which we refer from now on as *confidence*. We access all services and save the obtained raw results directly in the database for later analysis.

Finally, in context of the data collection, we issue 99,108 requests per service. As this exceeded the evaluation quota per service, we opt, where necessary and possible, for specific (academic) licenses.

Testing and Training Data: To obtain the results described in Section 4, we split the collected data, i.e., our corpus of 10,500 CAPTCHAs, into two disjunct sets: a training and a testing set. While the training set is used to create the meta-classifiers, the testing set is used for evaluating the ML services in the context of solving CAPTCHA challenges (*RQ1* – *RQ3*). We perform 10-fold cross-validation by randomly splitting the collected data into 10 subsamples with data equally distributed along the different categories. Out of the 10 subsamples, always one constitutes the testing data (10%) and the rest nine constitute the training data (90%). We made 10 passes in which we considered the first subsample as testing data, then the second subsample, then the third, etc. The results reported and discussed in Section 4 are the average of the results for each pass.

Data Analysis: To answer *RQ1*, we calculate the precision and recall based on the manually classified images, as a reference to the ground truth. On the other hand, to answer *RQ2* and *RQ3*, we need to calculate the accuracy of the ML services with respect to breaking the CAPTCHAs. Thus, we devise a method to compare the results of each service with the ground truth of previously manually labeled images. One key challenge is to build a meta-classifier that predicts, based on the results of a service, whether each individual image is a correct answer to its encompassing challenge or not. We build such a meta-classifier for each

service S and for each challenge category C. We implement each meta-classifier for $[S, C]$ following three main steps: (i) Collect all keywords K retrieved from S for images belonging to the challenges of C; (ii) For each keyword, find the confidence threshold T that yields the highest accuracy in predicting a correct (`TRUE`/`FALSE`) flag for an image; (iii) Find the best combination of $[K, T]$ pairs with the highest accuracy in predicting an accurate answer for a challenge.

At first, we collect all the keywords that are retrieved from S for all the images that belong to a challenge of category C. This is straightforward in all the services except for Microsoft's, whose response contains both a text with a confidence and a number of tags. Therefore, we have to select a different strategy for Microsoft's service. As such, we chose to extract the keywords for this service by tokenizing the text and omitting the tags.

Next, for each keyword K, we assume a confidence threshold T. We go through all images of C and corresponding responses from S and mark as true positive the cases where the image is manually labeled as correct answer to the challenge and (i) K is found by case-insensitive String matching in the response of S and (ii) the retrieved confidence is equal to or higher than T. Accordingly, we calculate the number of true negatives, false positives, and false negatives. We also calculate the per-case *accuracy* by diving the sum of true positives and true negatives to the total number of images. Technically, we start with a threshold value of 0 and increase it with a step of 1 up to 100. We apply the above process for each K in the $[S, C]$ pair. An example of the produced dataset is depicted in Table 1. As it can be seen, with increasing confidence thresholds values, accuracies decrease since the comparison becomes stricter. Having this dataset in place for each K, selecting the "best" confidence threshold is simply a matter of picking the one with the highest accuracy. In case two or more thresholds have the same accuracy (e.g., 48 – 50 in Table 1), we select the one with the highest value to avoid potential false positives. Finally, we create a list of all the keywords along with their best confidence thresholds and the accuracies that correspond to these thresholds; we sort the list by the accuracies.

Table 1. Excerpt from generated dataset for the keyword "automobile" for the AWS service and for the "Select all images with cars" challenge category.

Threshold	Confusion Matrix				Accuracy (%)
	TP	TN	FP	FN	
48	465	1627	3	677	75.46
49	465	1627	3	677	75.46
50	465	1627	3	677	75.46
51	462	1627	3	680	75.36
52	457	1627	3	685	75.18
53	451	1627	3	691	74.96

As the aforementioned example shows, trying to find the keyword "automobile", accompanied by a confidence threshold greater than or equal to 50%, in the response of the AWS service to an image of a "cars" challenge is a promising way of getting an accurate prediction on whether to select this image as an answer or not. However, will the prediction improve if we include more $[K, T]$ pairs? If so, what is the optimal number of pairs that should be included?

To investigate these questions, we calculate the number of challenges that would be accurately solved (without any mistake) when considering only the head of list. Specifically, we mark as `TRUE` the images whose responses from S (i) contain the keyword of the head of the list and (ii) the accompanying confidence is greater or equal to the confidence threshold of the head of the list. In our example, this case is when the response of the AWS service for images belonging to the "cars" category contains the keyword "automobile" with a confidence greater or equal to 50%. We then compare the marked images to the manually labeled ones—a match indicated an accurate solution of the challenge.

We repeat the above process by considering this time the first two items in the list, then the first three items, and so on. In the end, we are able to determine the $[K, T]$ pairs that yield the most accurate predictions—we call these pairs "optimal keywords" for the $[S, C]$ pair.

4 Evaluation

In this section, we present the results of our study and structure them according to the three research questions. In detail, for each question, we first report on the results and then provide a preliminary (subjective) interpretation. Prior to that, we give an overview of the datasets and services we used in our study.

4.1 Datasets and Services Used

Our corpus consists of 10,500 CAPTCHAs belonging to seven different categories. Each category corresponds to the original prompt of the challenge such as "Select all images with house/-store front/street number". The distribution of CAPTCHAs per categories is depicted in Figure 2. As can be seen, the two most popular categories are *Store Front* and *Street Number*, which occupy 40% and 34% of the study data, respectively. The smallest category, *Road*, amounts to 2% of the study data, in particular to 192 CAPTCHAs. Each CAPTCHA contains 8, 9, or 16 images (CAPTCHA *size*) arranged in a grid of 2x4, 3x3 and 4x4, respectively. CAPTCHAs of different sizes are not uniformly distributed in the categories. Instead, 16-sized CAPTCHAs belong exclusively to the *Street Sign* category, 8-sized ones belong exclusively to the *Store Front* category,

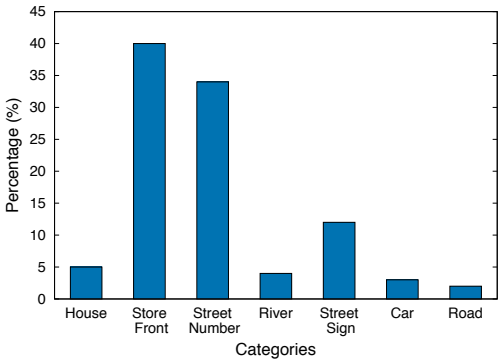

Fig. 2. Distribution (in percentage) of collected CAPTCHAs in categories.

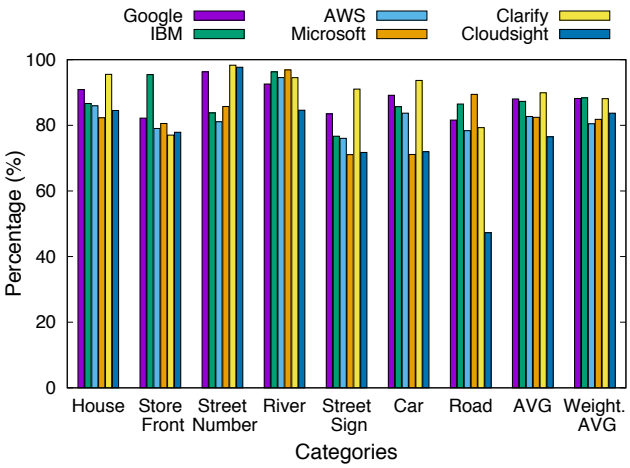

Fig. 3. Precision of ML services in classifying an image as a correct answer to the encompassing CAPTCHA challenge.

and 9-sized ones belong exclusively to the one of the other five categories. We discuss how the different per-category sizes may have influenced the results of our study in the next sections. It is worth to be mentioned that we could have normalized the results by artificially enforcing size 8 for all categories, however we chose to preserve the original CAPTCHA sizes in order to be able to draw valid conclusions on the ability to break the original CAPTCHAs. The total number of images we collected and processed was 99,108 (on average 9.44 images per CAPTCHA).

4.2 RQ1: Precision and Recall of ML Services

To investigate the differences between the performance of ML services in the different categories, we calculate the precision and recall of each meta-classifier that corresponds to a service-category pair across all images contained in all CAPTCHAs of the category. The results are illustrated in Figures 3 and 4.

A first observation from Figure 3 is that, with a single exception that of Cloudsight in *Road*, all services yield a precision higher than 70% in all categories, while the best precision in all categories is higher than 90%. On average, the best precision (irrespective of the service providing it) is 94%.

Looking at the results of the services across all categories (two rightmost groups in the Figure 3), *average* is the mean value of calculated per-category precisions, while *weighted average* is the precision calculated on the total number of images, irrespective of the category of their encompassing CAPTCHAs. Since some categories contain more CAPTCHAs, these two values are different for each service, with the weighted average "boosting" the services which score higher

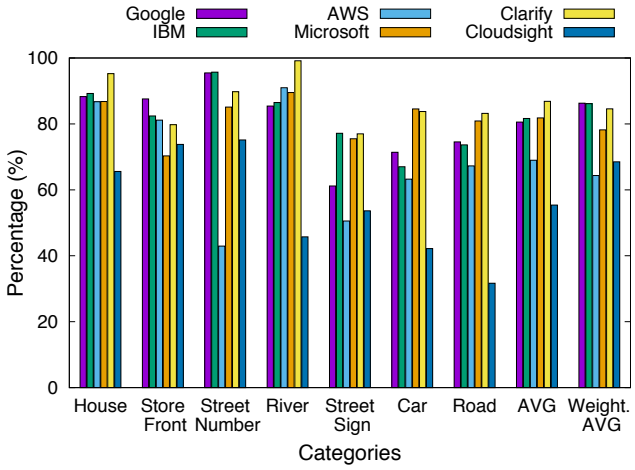

Fig. 4. Recall of ML services in classifying an image as a correct answer to the encompassing CAPTCHA challenge.

in the popular categories of *Store Front* and *Street Number*. Yet, both statistics yield values between 76% and 89%, with small variations among the services.

Figure 4 depicts the recalls of our meta-classifiers corresponding to each service-category pair. The best recall per category (irrespective of the service providing it) ranges from 77% to 99% with an average value of 89%. The categories with the highest recalls are *House*, *Street Number*, and *River*, while the one with the lowest recalls is *Street Sign*. With respect to the services, Cloudsight scores consistently low in all categories (an average of 55%). The rest of the services score consistently high (more than 80% on average), with the exception of AWS which scores a mere 43% in *Street Number* and 50% in *Street Sign* (and obtains an average recall of 69%).

Interpretation: Our meta-classifiers yield a consistently high precision. As for the recall, the low values of *Street Sign* can be attributed to the following reason. Images of street signs, contrary to images of other categories such as cars, are usually fragmented across several individual images. Human cognition should be able to easily identify fragments of street signs by imagining the missing parts; it seems that this is challenging for ML algorithms, which miss a number of correct responses (an increase in false negatives).

4.3 RQ2: Absolute Accuracy of ML Services

Absolute accuracy is the case in which we try to solve CAPTCHAs without tolerating a single mistake in the binary classification (*selected* or *not selected*) of the images included in each CAPTCHA. Figure 5 depicts the results for this case.

Similar to the precision and recall case, the second group from the right, *average*, is the mean value of calculated per-category accuracies. The right-

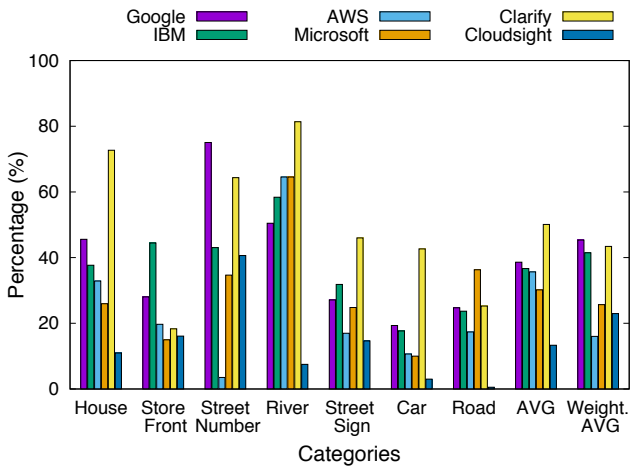

Fig. 5. Absolute accuracy (in percentage) of ML services: solving CAPTCHAs without any mistake tolerance.

most group, *weighted average*, is the accuracy calculated on the total number of CAPTCHAs, irrespective of their category.

The results indicate that for each category there is at least one service that scores higher than 35%, with *House*, *Street Number* and *River*, having services that score up to 72%, 75%, and 81%, respectively. Overall, the services score lower in *Store Front*, *Street Sign*, *Car*, and *Road* and higher in *River*. There are also many differences in the performance of the services in different categories. For example, Microsoft and Google provide the most accurate services for *Road* and *Street number*, respectively, while Clarifai is the most accurate in *House*, *River*, *Street Sign*, and *Car*. AWS scores comparatively high in *House* and *River*, but extremely low (3%) in *Street Number*. At any rate, on average, Clarifai and Google, closely followed by IBM and AWS, are the most accurate services, with an average accuracy of close to 40%.

Interpretation: With the exception of Clarifai and Cloudsight, which are scoring consistently high and low respectively, the high variation in the results of the other services can be attributed to the difference in the datasets used in the training of their internal ML classifiers. The reason why some categories yield lower accuracy could be as follows. A street number, although blurry, is entirely contained in an image, while a river can be recognized by its characteristic shape and color. A street sign, however, does not have any characteristic color and, as explained also in the case of recall, its characteristic shape is often not identifiable as it is typically not entirely contained in an image. Further, the low accuracy in the *Street Sign* category should also be attributed to the larger CAPTCHA size (16 images per CAPTCHA).

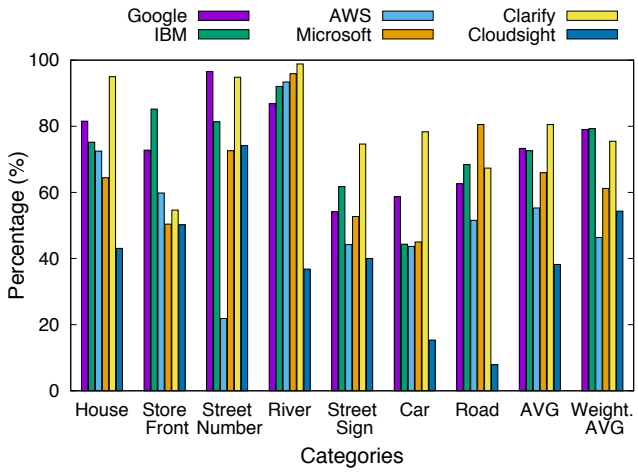

Fig. 6. Sufficient accuracy (in percentage) of ML services: solving CAPTCHAs by tolerating one mistake.

Finally, although the average accuracy of services is not high, note that the absolute accuracy test is also challenging for humans. That might be also the reason why CAPTCHA services such as reCAPTCHA typically allow for one mistake per challenge. In the following, we report our results for this very case.

4.4 RQ3: Sufficient Accuracy of ML Services

Sufficient accuracy describes the case in which we try to solve CAPTCHAs by tolerating one mistake in the binary classification of images included in each CAPTCHA. Figure 6 shows the results for this case.

A first observation is that for each category there is at least one service with a sufficient accuracy of 75% or more; for three categories, the best accuracy is even 95% or more (River, Street Number, Road). On average, the best accuracy (irrespective of the service providing it) is 87%. Similarly to the analysis of the absolute accuracy, *Street Sign* and *Car* are the worst performing categories. We can also observe sharp differences in the performance of services in some categories: AWS scores a mere 22% in *Street Number*, where all other services score more than 70%. In the categories *House* and *Store Front*, the services of Clarifai and of Microsoft stand out as far better than the other services scoring accuracies of 95% and 85%, respectively.

The two rightmost groups have been calculated as it was the case for the absolute accuracy. It seems clear that, on average, the three most accurate services are the ones offered by IBM, Clarifai, and Google with a weighted average accuracy 79%, 79%, and 75% respectively. Microsoft scores a weighted average accuracy of 61%, while Cloudsight and AWS score 54% and 46%, respectively.

Finally, comparing the results of the sufficient accuracy case with the absolute accuracy, we can observe that in most categories the relative difference between the services accuracies is preserved, while the absolute values have strongly increased. When looking at the best performing services per category, their accuracy across the two cases is increased on average by 30 percentage units, with a minimum increase of 17 units (Clarify for *River*) and a maximum of 44 units (Microsoft for *Road*).

Interpretation: Similarly to the absolute accuracy case, the sharp variations in the performance of services across different categories can be explained by different datasets used in the training of the internal ML classifiers, while differences between categories can be attributed to both the object containment case as well as the CAPTCHA size.

5 Security Impact and Implications

The main question that still remains is: Assuming that CAPTCHA systems are widely used to tell humans from computers apart, are these systems vulnerable to attacks that utilize modern image recognition services?

5.1 Automated CAPTCHA Solver

To investigate the above question, we implemented an automated program (bot) which visits websites that contain CAPTCHAs and attempts to automatically break them, without any human interaction. To do so, the bot performs the following steps:

Step 1: Visit a website, find and retrieve the `iframe` which contains the checkbox CAPTCHA, and automatically click it.
Step 2: Extract the individual images contained in the CAPTCHA challenge. If we have low confidence for this challenge category or if this is a completely new category for which we do not have any information, press the reload button at the bottom left of the challenge and retrieve a new challenge (presumably of different category). If such a case occurs, we additionally store the new images for further advancing our prediction model.
Step 3: Submit the extracted images to the image recognition service with the most promising results for the particular challenge category and retrieve the results (keywords and confidence).
Step 4: Predict whether each image is a correct answer to the challenge using our meta-classifier for the service-category pair. In particular, check whether at least one keyword from the service is included in the optimal keywords of the meta-classifier and the confidence of the included keyword is higher than the optimal threshold.
Step 5: Use the predictions from the meta-classifier and based on that click accordingly.

Step 6: Press the `VERIFY` button on the bottom right corner. In case the challenge is correctly solved, the bot retrieves an *"I'm not a robot"* response.

We implemented the above process with Selenium, a browser automation framework, which is able to render the DOM of a web page, execute JavaScript, and handle keyboard and mouse events. We implemented the bot in Python; it consists of less than 200 lines of code. In the rest, we elaborate on the selection of the optimal service per category as well as the economic viability of our solver.

Optimal Service for Category: Based on our study results, there is not a single service to rule them all. However, for each CAPTCHA category, there exists a service that yields the most promising results, measured by the accuracy of its meta-classifier described above. In particular, based on the sufficient accuracy results reported in Section 4.4, we created a mapping between categories and services. When this metric had the same value for more than one services ($Clarifai$ and $Microsoft$ in *river*), we looked into the absolute accuracy results to determine the optimal service (in this case, $Clarifai$). Table 2 provides an overview of the optimal service per CAPTCHA category. When considering the optimal categories, our prototype bot implementation yielded fruitful results in the sense of breaking the CAPTCHAs with the sufficient accuracies described above.

Table 2. Service selection per CAPTCHA category.

Challenge	Service
House	Clarifai
Store front	IBM
Street number	Google
River	Clarifai
Street sign	Clarifai
Car	Clarifai
Road	Microsoft

Economic Feasibility: Since CAPTCHA solving is offered as a paid service in the underground economy, it is worthwhile briefly assessing also the economic feasibility of our approach when using an automated solver. All services employ a pay-as-you-go model and most of them charge a similar amount of money per request (i.e., for the labeling of a single image). The time this work took place, the services considered in our study, except Cloudsight, charge from $1 to $2 per 1000 requests. Cloudsight charges a higher amount, about $50 per 1000 requests. One reason behind this might be that Cloudsight relies not only on ML algorithms, but also on human labor for image labeling as part of crowdsourcing. Nevertheless, since Cloudsight does not appear in the optimal services group shown in the section before, the operational cost of our CAPTCHA solver, assuming an average of 10 images per CAPTCHA, would be $0.01 to $0.02.

5.2 What Does This Mean for the Future of CAPTCHAs?

There exist currently millions of websites leveraging CAPTCHAs as protection mechanism against web spam, online attacks, and automated scripts. The most resilient type of CAPTCHAs has, so far, been image-based CAPTCHAs. At the same time, the most widely used implementation of image-based CAPTCHAs is embodied in Google's reCAPTCHA service.

Our work shows that it is possible to create an automated solver for reCAPTCHA, notably *without being a machine learning expert, without having access to a large*

corpus of images, or setting up and operating any ML infrastructures. In fact, invoking publicly available services following a pay-as-you-go model would even be feasible from an economic (underground) perspective as shown above.

Our approach relies on a small upfront effort in manually solving a sufficient number of challenges from each CAPTCHA category. As an indicator for the necessary effort, the smallest training set we used (for *Road* category) contained only 172 CAPTCHAs and provided already an average sufficient accuracy of 56% across all services. It is reasonable to expect that this will only increase by increasing the number of samples for this category. Since reCAPTCHA contains only a limited number of categories (during our study, we encountered 16 categories, out of which we selected the first seven w.r.t number of samples for inclusion), the manual effort required for keeping the solver up-to-date is not high. After manually solving a number of challenges per category, the operation of the solver is fully automated. Importantly, *its expected accuracy is as high as 88%* (average of best sufficient accuracies per category from Section 4.4).

From the security perspective, the creation of the automated CAPTCHA solver signifies a successful generic ML-based attack. Since our attack is based on Cloud services that are expected to continue to be available, appropriate countermeasures should be taken to prevent similar attacks in the future. One possibility lies in distorting the images so that the ML cannot recognize the objects anymore with high confidence. The limitation here is, of course, that the images have to still be recognizable by humans; unfortunately, there is not much room for further blurring or distorting the reCAPTCHA images for humans. Another possibility lies in increasing the semantic information necessary to solve the challenge. Consider, for instance, requesting the user to identify a combination of specific items or actions, e.g., groups of two to three persons drinking beer. Finally, CAPTCHAs could rely more on fragmented views of a scene, such as the *Street Sign* CAPTCHAs in our study; we observed that these pose a challenge to ML (low accuracies) but not at all to humans.

In any case, and given the current state of using image-based CAPTCHAs as a security measure, we conclude that they are insufficient to be kept as the de-facto standard to prevent automated security attacks.

6 Threats to Validity

The presented study is a curiosity-driven study with many manual and automated tasks. Inherent to such tasks are a number of threats to validity out of which we now discuss those that appear to be major ones from our own perspective. One major threat to validity concerns the trustworthiness of the used oracle (ground truth) to train and test the ML services. This ground truth was defined manually by solving all images and, thus, it affects the internal validity of the whole study. We tried to minimize the threat by defining the ground truth dataset in pairs of researchers. Still, we cannot guarantee that we did not wrongly classify some of the images even though we postulate that it would negatively affect the accuracy of all the ML services probably in the same way.

Another threat to validity arises from the fact that we do not know the extent to which the source of the images matter. We took all images from Google to train all ML services, but argue that the choice of images seems to have a lower impact as (*i*) they resemble regular photos of real-life situations and (*ii*) the results do not indicate to perpetually better scores by Google's ML service. To the best of our knowledge, we see no clear indicator that the choice of images for the training and testing dataset has influenced the outcome of the study.

Finally, another threat to validity concerns the external validity and eventually the conclusion validity. Are we able to draw conclusions that go beyond the image-recognition services? For instance, can we draw conclusions on the general field of machine learning? Please note that, again, our study was a curiosity-driven one and we deliberately (and also opportunistically) chose the services described in the paper. Although our intention was not to draw any conclusions beyond the selected services, we can still argue that the effects observed in the setting described in this paper could be observed in different settings.

7 Related Work

Since the concept of CAPTCHA was first introduced, a lot of research has been done in this area to create CAPTCHAs that are easy for humans to solve, yet extremely difficult for machines. The most popular category, since recently, was the text-based CAPTCHAs. However, modern *Optical Character Recognition* (OCR) algorithms were able to solve the presented challenges with pretty high accuracy [5, 39, 40]. The suitability of CAPTCHAs as a means to implement Turing Tests in a usable manner has since then been discussed for some years now. In particular, Bursztein et al. [3] introduced a novel approach to solving CAPTCHAs in a single step that uses machine learning to attack the segmentation and the recognition problems simultaneously. Baecher et al. [2] analyzed three recent generations of reCAPTCHA and presented an algorithm that is capable of solving at least 5% of the challenges generated by these versions. Cruz-Perez et al. [11] presented a novel approach for automatic segmentation and recognition of reCAPTCHA in websites which is based on CAPTCHA image preprocessing with character alignment, morphological segmentation with three-color bar character encoding, and heuristic recognition.

Therefore, alternatives to text-based CAPTCHAs were considered a necessity. Goswami et al. [15] presented FaceDCAPTCHA, a face detection-based CAPTCHA, in which four to six distorted face/non-face images are embedded in a complex background and a user has to correctly mark the center of all the face images within a defined tolerance. Another alternative is the video-based CAPTCHA challenges such as NuCAPTCHA. However, Xu et al. [38] presented flaws in the design of video-based CAPTCHAs by implementing automated attacks based on computer vision techniques as a proof of concept.

Another category of CAPTCHAs discussed in literature is based on audio. Nevertheless, researchers, once again, were able to break these challenges. For instance, Kopp et al. [25], pointed out flaws and weak spots of frequently used

solutions and concluded with consequent security risks. Meutzner et al. [27] suggested to use speech recognition rather than generic classification methods for better analyzing the security of audio-based reCAPTCHAs. They showed that their attack, based on an automatic speech recognition system, can defeat reCAPTCHA with a significantly higher success rate than reported in previous studies.

The closest work to our study is the one from Sivakorn et al. [34]. The authors propose an attack that uses deep learning technologies to annotate images. They focus, on trying to automatically break reCAPTCHA challenges and succeed in roughly 70% of the cases. In particular, they also used Clarifai as their main service to break a CAPTCHA. However, as we have shown in our work, Clarifai did not perform equally good in all of the reCAPTCHA challenges. In fact, although the scope of their work differs in the sense of providing an (also technical) analysis of CAPTCHAs, their work inspired some technicalities in our own study design which aims at providing a broader analysis of the suitability of publicly available ML services to break CAPTCHAs and the security implications this has.

Compared to previous works that attempt to break CAPTCHAs by implementing a specific approach or algorithm, we took a different path. The motivating question is to understand the extent to which publicly available ML services can pass image-based Turing Tests and the security implications this has. We are, thus, leveraging working solutions, which are offered in the form of online image recognition services, and apply them for CAPTCHA infiltration. Another key difference with the previous works is that the sole goal of our study is not to break the CAPTCHA mechanism, but to compare existing services and evaluate their ability to extract valuable and accurate knowledge from an image. We only use the CAPTCHA mechanism as benchmark for our comparison. Yet, we also saw that the overall image recognition technology has advanced to the point that can be used for malicious purposes as well. To the best of our knowledge, we are the first to perform such a comparison among different image recognition services.

8 Conclusion

In this study, we wanted to understand the extent to which today's publicly available ML services can be used to pass image-based Turing Tests. Thus, we employed six ML services to a broad set of available CAPTCHAs. Our results strengthened our confidence in the suitability of available ML services to break CAPTCHAs and pose a security threat if used automatically. Interestingly, it was possible to create an automated solver for reCAPTCHAs, notably without prior expertise in machine learning, without having build up an own large corpus of images, and without setting up and operating specific ML infrastructures on our own. This manifests the idea that today's available and ready-to-use ML services can indeed be used to pass image-based Turing Tests and rises new questions to the security of systems that rely on this technology as a security measure.

Acknowledgments

This work was supported by the European Union's Horizon 2020 research and innovation programme under grant agreement No. 833115 (PREVISION).

References

1. Amazon Rekognition. Deep Learning-Based Image Recognition — Search, Verify, and Organize Millions of Images. https://aws.amazon.com/rekognition/.
2. P. Baecher, N. Büscher, M. Fischlin, and B. Milde. Breaking reCAPTCHA: A Holistic Approach via Shape Recognition. In *Future Challenges in Security and Privacy for Academia and Industry*, 2011.
3. E. Bursztein, J. Aigrain, A. Moscicki, and J. C. Mitchell. The End Is Nigh: Generic Solving of Text-Based CAPTCHAs. In *USENIX Workshop on Offensive Technologies (WOOT)*, 2014.
4. E. Bursztein, R. Beauxis, H. Paskov, D. Perito, C. Fabry, and J. Mitchell. The Failure of Noise-Based Non-Continuous Audio Captchas. In *IEEE Symposium on Security and Privacy*, 2011.
5. K. Chellapilla and P. Y. Simard. Using Machine Learning to Break Visual Human Interaction Proofs (HIPs). In *Advances in neural information processing systems*, 2005.
6. F. Chollet. *Deep Learning With Python*. Manning, 2017.
7. Clarifai. Artificial Intelligence With a Vision. https://clarifai.com/.
8. Cliq Orange. https://www.cliqorange.com/.
9. Cloudsight. Visual Cognition — High Quality Understanding of Images Within Secconds. https://cloudsight.ai/.
10. R. Collobert and J. Weston. A Unified Architecture for Natural Language Processing: Deep Neural Networks With Multitask Learning. In *International conference on Machine learning*, 2008.
11. C. Cruz-Perez, O. Starostenko, F. Uceda-Ponga, V. Alarcon-Aquino, and L. Reyes-Cabrera. Breaking reCAPTCHAs With Unpredictable Collapse: Heuristic Character Segmentation and Recognition. In *Pattern Recognition*, 2012.
12. L. Deng and D. Yu. Deep Learning: Methods and Applications. *Foundations and Trends in Signal Processing*, 7(3–4):197–387, 2014.
13. Google. reCAPTCHA: Protect Your Site From Spam and Abuse. https://developers.google.com/recaptcha/.
14. Google Cloud Vision. Derive Insight From Images With Our Powerful Cloud Vision API, https://cloud.google.com/vision/ 2017.
15. G. Goswami, B. M. Powell, M. Vatsa, R. Singh, and A. Noore. FaceDCAPTCHA: Face Detection Based Color Image CAPTCHA. *Future Generation Computer Systems*, 31:59–68, 2014.
16. Henrik de Gyor. *Keywording Now: Practical Advice on Using Image Recognition and Keywording Services*. Another DAM Consultancy, 2017.
17. G. Hinton, L. Deng, D. Yu, G. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, B. Kingsbury, et al. Deep Neural Networks for Acoustic Modeling in Speech Recognition. *IEEE Signal processing magazine*, 29, 2012.
18. hive.ai. Powering Artificial Intelligence. https://thehive.ai/.
19. IBM. Watson Visual Recognition. https://www.ibm.com/watson/services/visual-recognition/, 2017.

20. ICrimson Hexagon. https://www.crimsonhexagon.com/.
21. Imagga. Build Your Apps on Top of an Advanced Image Tagging Technology. https://imagga.com/.
22. Jastec. A Pioneer of Image Recognition. http://www.jastec.fr/.
23. B. Kolosnjaji, G. Eraisha, G. Webster, A. Zarras, and C. Eckert. Empowering convolutional networks for malware classification and analysis. In *International Joint Conference on Neural Networks (IJCNN)*, pages 3838–3845. IEEE, 2017.
24. B. Kolosnjaji, A. Zarras, G. Webster, and C. Eckert. Deep Learning for Classification of Malware System Call Sequences. In *Australasian Joint Conference on Artificial Intelligence*, 2016.
25. M. Kopp, M. Pistora, and M. Holena. How to Mimic Humans, Guide for Computers. In *ITAT*, 2016.
26. Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-Based Learning Applied to Document Recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.
27. H. Meutzner, V.-H. Nguyen, T. Holz, and D. Kolossa. Using Automatic Speech Recognition for Attacking Acoustic CAPTCHAs: The Trade-Off Between Usability and Security. In *Annual Computer Security Applications Conference (ACSAC)*, 2014.
28. Microsoft. Computer Vision API. https://azure.microsoft.com/en-us/services/cognitive-services/computer-vision/.
29. NY Times. Please Prove You're Not a Robot. https://www.nytimes.com/2017/07/15/opinion/sunday/please-prove-youre-not-a-robot.html.
30. E. Reshef, G. Raanan, and E. Solan. Method and System for Discriminating a Human Action From a Computerized Action, 2004.
31. O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.
32. Saltlab World. Image & Object Recognition System & API. http://saltlabworld.com/.
33. Scale. Image Annotation API. https://www.scaleapi.com/image-annotation.
34. S. Sivakorn, I. Polakis, and A. D. Keromytis. I Am Robot: (Deep) Learning to Break Semantic Image CAPTCHAs. In *IEEE European Symposium on Security and Privacy (EuroS&P)*, 2016.
35. A. M. Turing. Computing Machinery and Intelligence. *Mind*, 59(236):433–460, 1950.
36. Vize.ai. Custom Image Recognition API. CustomimagerecognitionAPI.
37. L. Von Ahn, M. Blum, N. J. Hopper, and J. Langford. CAPTCHA: Using Hard AI Problems for Security. In *European Cryptology Conference (EUROCRYPT)*, 2003.
38. Y. Xu, G. Reynaga, S. Chiasson, J.-M. Frahm, F. Monrose, and P. C. van Oorschot. Security and Usability Challenges of Moving-Object CAPTCHAs: Decoding Codewords in Motion. In *USENIX Security Symposium*, 2012.
39. J. Yan and A. S. El Ahmad. Breaking Visual Captchas With Naive Pattern Recognition Algorithms. In *Annual Computer Security Applications Conference (ACSAC)*, 2007.
40. J. Yan and A. S. El Ahmad. A Low-Cost Attack on a Microsoft CAPTCHA. In *ACM Conference on Computer and Communications Security (CCS)*, 2008.