Breaking Google reCaptcha v2 with Transfer Learning

Marco Aurélio Beber and Yuri Santa Rosa Nassar dos Santos
Department of Informatics and Statistics
Federal University of Santa Catarina, Florianópolis, Brazil

1. INTRODUCTION

Since the creation of the WWW the amount of data and information has exponentially increased. Web scraping emerged to automate the process of data transfer between a web page and a structured data format **[1]**. As we needed to automate data collection, new technologies were developed to protect web pages from these automated processes. In agreement with that, Ahn et al. **[2]** defined the captcha method to verify human users through a cognitive process that computers cannot imitate. Moreover, modern Captchas, such as Google reCaptcha, use image puzzle challenges **[3]**.

On the other hand, there are some services that solve reCaptchas v2 at the cost of US$ 2.99 per 1,000 challenges, implying significant costs to companies that rely on web scraping. Recently, transfer learning emerged as a viable solution for breaking some types of reCaptcha challenges **[4]**. Based on recent work, we propose to use transfer learning from three well established architectures (VGG19 **[5]**, ResNet50 **[6]** and InceptionV3 **[7]**) pretrained on Imagenet to break the two simplest challenges of reCaptcha v2.

The remainder of this poster is as follows. In Section 2, we describe the method. Section 3 describes the experiments performed to choose the best architecture and configurations for each type of query. Finally, in Section 4, we summarize our results and offer conclusions.

2. METHOD

In this section we present the method used to break reCaptcha challenges, as illustrated in Figure 1. Our dataset is composed of two different types of challenge: (i) image select challenges, which consist of selecting all individual images that match a given query; and (ii) tile select challenges, which consist of selecting parts of a single image that match a given query.

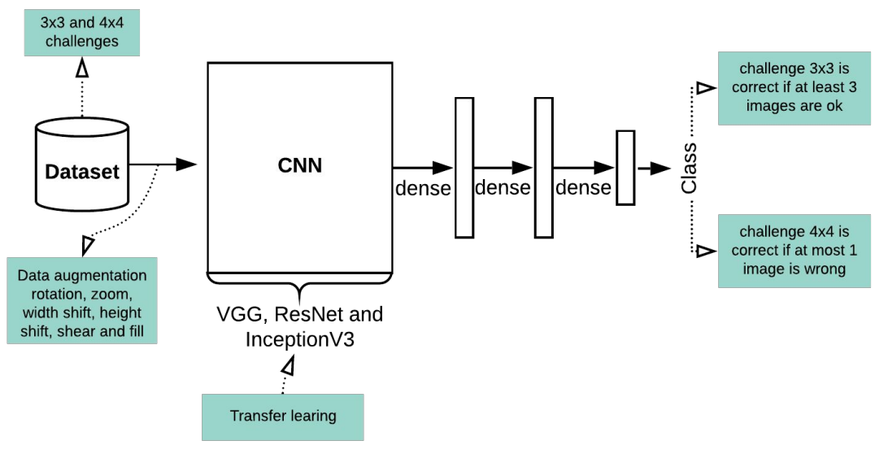

Figure 1. Method adopted in the experiments

For each type of challenge there are multiple types of queries available. Since there are multiple types of queries with few examples, we focus on four types of queries that correspond to up to 65% of the whole dataset: (i) traffic lights; (ii) cars; (iii) palm trees; and (iv) bridges.

As we have two types of challenge with images of different sizes, we preprocess the data accordingly: (i) image select challenges consist of 9 rgb images of size 100x100x3. We pad 12 pixels at the end of each image, resulting in 9 rgb images of size 112x112x3; (ii) tile select challenges consist of 1 rgb image of size 450x450x3. We discard the last 2 pixels and split the image into 16 images of size 112x112x3.

We train different binary classifiers for each type of query, experimenting with different network architectures and forms of transfer learning, choosing the best network using grid search on a validation set based on the macro average of the f-measure score. The training images are randomly augmented.

We further define a metric to measure the number of correct challenges. For image select challenges, the challenge is considered correct if at least 3 images of the given query are selected. As for tile select challenges, the challenge is considered correct if all but 1 image is correct.

Finally, we report the number of correct challenges considering the best model for each type of query and use the query distribution to predict how much money the models could save per 1,000,000 challenges.

3. EXPERIMENTS

In this section we evaluate our method with a manually collected reCaptcha v2 dataset. The dataset consists of 2,000 challenges, divided into 1,223 image select challenges, and 777 tile select challenges. As there are multiple types of queries with few examples, we perform experiments only on four types of queries: (i) traffic lights; (ii) cars; (iii) palm trees; and (iv) bridges, resulting in a dataset of 1,315 challenges. This dataset is split into train, validation, and test sets with the proportion of 70%/15%/15% respectively.

In our experiments, we consider three types of CNN architectures: (i) vgg19; (ii) resnet50; and (iii) inceptionV3. We also compare a baseline (network trained from scratch), with two different forms of transfer learning: (i) feature extraction, where the convolutional layers of the network are frozen; and (ii) fine tuning, where no layer is frozen, but the network is initialized with pre-trained weights. At the end of each convolutional network, we drop the existing fully connected layers and append two fully connected layers and a layer for classification. We treat each query type as a binary classification problem, where the true examples are the selected images or tiles, and the false examples are the remaining images or tiles in each challenge.

We perform grid search on the validation set based on the macro average f-measure score, for each type of query, for each architecture, for each method, varying the number of neurons at each fully connected layer in range [128, 256, 512, 1024], and varying dropout after each dense layer in range [0.3, 0.5], resulting in 288 variations.

We set the batch size = 64, the learning rate = 0.001, and train the networks with Adam for at most 50 epochs. The training set is randomly augmented with rotation [0, 15], zoom [0, 0.15], width and height shift [0, 0.15] and shear [0, 0.15].

We report the macro average f-measure for train and test sets in Table 1, where we highlight the best model for each query type.

Table 1. Best results between Transfer learning x CNN

Query type	Dataset	Transfer Learning - Feature Extraction			Transfer Learning - Fine Tuning			Baseline - Trained from Scratch		
		VGG	ResNet	Incept.	VGG	ResNet	Incept.	VGG	ResNet	Incept.
Traffic Lights	Train	0.96	0.95	0.90	0.94	0.94	0.93	0.93	0.93	0.92
	Test	0.87	**0.91**	0.84	0.89	0.89	0.89	0.89	0.90	0.89
Cars	Train	0.95	0.98	0.9	0.97	0.97	0.98	0.93	0.95	0.95
	Test	0.91	0.91	0.89	0.91	0.88	**0.92**	0.87	0.88	0.89
Palm Trees	Train	0.98	0.99	0.95	0.97	0.98	0.96	0.96	0.96	0.94
	Test	**0.96**	**0.96**	0.93	0.95	**0.96**	0.94	**0.96**	0.93	0.94
Bridges	Train	0.99	0.98	0.92	0.93	0.97	0.97	0.93	0.97	0.98
	Test	**0.95**	0.90	0.92	0.92	0.92	0.93	0.91	0.90	0.87

As seen in Table 1, transfer learning approaches consistently outperform the baseline. Only in query type **Palm Trees** the baseline achieved the same result as the other two transfer learning methods. Regarding the transfer learning approach, feature extraction shows better performance over fine tuning, as it achieved the best results for three types of query. As for the network architecture, it is not possible to say which is better, as each architecture excels in a different type of query.

Furthermore, we report the best network configurations for each type of query/challenge along with the number of correct challenges computed from the test set. We also present the challenge distribution from our dataset to predict how much our models could save per 1,000,000 challenges.

Table 2. Best configuration for the top models

Query type		Best model			
		Configuration	% Correct challenges	Challenge distribution	Savings (US$) for each 1,000,000 reCaptchas
Traffic Lights	3x3	**Architecture:** ResNet **Method:** Feature Extraction **Neurons**: 256 **Dropout:** 0.5	0.8	0.10	$ 239.20
	4x4		0.75	0.24	$ 538.20
Cars	3x3	**Architecture:** Inception **Method:** Fine Tuning **Neurons**: 1024 **Dropout:** 0.3	0.7878	0.16	$ 376.88
Palm Trees	3x3	**Architecture:** [VGG, ResNet, ResNet, VGG] **Method:** [Feature Extraction, Feature Extraction, Feature Extraction, Trained from Scratch] **Neurons:** [128, 256, 512, 1024] **Dropout:** [0.5, 0.3, 0.3, 0.5]	0.9375 1 1 0.875	0.08	$ 224.20 $ 239.20 $ 239.20 $ 209.30
Bridges	3x3	**Architecture:** VGG **Method:** Feature Extraction **Neurons:** 128 **Dropout:** 0.5	0.9333	0.07	$ 194.30

As seen in Table 2, the four challenges correspond to up to 65% of the original dataset. Considering the cost of $ 2.99 per 1,000 challenges, the distribution of each challenge and the % of correct challenges, we end up with savings of $ 1,587.78 per 1,000,000 challenges, which correspond to savings of 53.10%.

4. CONCLUSION

In this poster we presented a method for breaking google reCaptcha v2 using transfer learning. Indeed, we showed that transfer learning approaches surpassed the defined baseline for our dataset. Furthermore, transfer learning methods showed strong results (> 90% average f-measure) for each binary classification problem. Considering the challenges, our method could save costs up to 53.10% by only using 65% of all available challenges. Overall, we showed that it is possible to break the reCaptcha v2 challenge using transfer learning even with a small dataset of 2,000 instances. As future work, we intend to collect more data and to consider more types of queries into our solution.

REFERENCES

[1] Gheorghe, M., Mihai, F., & Dârdală, M. (2018). Modern techniques of web scraping for data scientists. *Romanian Journal of Human - Computer Interaction, 11*(1), 63-75.

[2] Luis von Ahn, Manuel Blum, Nicholas J. Hopper, and John Langford. (2003). CAPTCHA: Using hard AI problems for security. *International Conference on the Theory and Applications of Cryptographic Techniques*, Warsaw, Poland, pages 294–311. Springer

[3] Csuka, K., Gaastra, D., & de Bruijn, Y. (2018). Breaking CAPTCHAs on the Dark Web.

[4] Yuan Zhou, Zesun Yang, Chenxu Wang, and Matthew Boutell. (2018). Breaking google reCaptcha V2. *J. Comput. Sci. Coll.* 34, pages 126-136.

[5] Simonyan, K., Zisserman, A., (2014). Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv technical report*.

[6] K. He, X. Zhang, S. Ren and J. Sun. (2016). Deep Residual Learning for Image Recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, pp. 770-778.

[7] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens and Z. Wojna. (2016). Rethinking the Inception Architecture for Computer Vision. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, pages 2818-2826.