It's A Cat or A Dog: Defenses Against Machine Learning Attacks on Image Captcha

Pranav Panage
University of Georgia
Athens, Georgia
Email: pranav.panage@uga.edu

***Abstract*—Traditional Captchas consisting of letters and numbers are not really user friendly. In recent years, big web technology companies like google, facebook, etc. have opted for object recognition based Captchas. But advancement in recent years in the field of Machine Learning has made these object recognition Captchas as well as traditional Captchas vulnerable. In this paper we attempt to defend the Machine Learning attacks against object recognition Captcha with a Concept Drift. We aimed at reducing the accuracy of these machine learning technique by systematically introducing the noise in Image Captcha. In the evaluation we test our defenses against SVM classifier. We also evaluate various state of the art image recognition API's and how they stack up against our counter-attack.**

I. Introduction

According to Wikipedia, A CAPTCHA (a backronym for "Completely Automated Public Turing test to tell Computers and Humans Apart") is a type of challenge-response test used in computing to determine whether or not the user is human[2]. The Traditional captcha's ask user to identify letters and/or numbers in image. These letter are often in distorted arrangement. This distortion makes sure that machines are not able to identify letters and numbers from these captcha's. In process of making it difficult to machines it also makes it difficult for humans to identify the correct character sequence. On average humans take 10 seconds to correctly solve typical captcha.

Considering these drawbacks in 2007 Microsoft came up with Object Recognition Image Based captcha called ASIRRA[3]. In this captcha users were asked to identify if the animal in the image is cat or dog. This captcha had over three million labeled images from Petfinder.com. Users are shown 12 images and then they are asked to identify cat images out of those. Their user study has shown that humans can do this task with an accuracy of 99.6% [4].Also these tasks are more fun to do compared to traditional cumbersome captcha's. This was a promising approach towards captcha.

But in 2008 Golle et al.[1] published a paper "Machine Learning Attacks Against the Asirra Captcha". This paper describes a SVM based classifier where authors were able to achieve the 82.7% accuracy, which translates to 10.3% probability of identifying all 12 images correctly. Later on in October 2014 ASIRRA project was closed [5]. But that wasn't the end of Image Captcha. Hereafter in this paper we will refer object recognition based Image captcha as Image Captcha. Figure 1 shows typical structure of ASIRRA

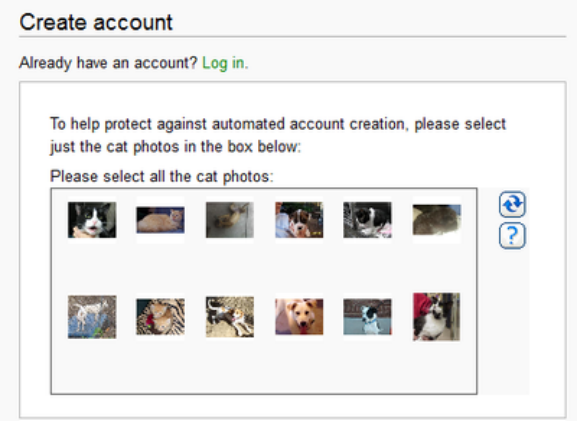

Fig. 1. An Asirra challenge. The user selects each of the 12 images that depict cats.

captcha. These machine learning attacks did not stop others from implementing image captcha. Google's reCaptcha[6] and facebook are still using Image Captcha.

In this paper we aimed to reduce the accuracy of machine learning attacks against these Image Captcha. We have added various types of noises to these images which we will see in details in Section 3. We were able to successfully reduce the accuracy of these classifiers. We have tested our noisy images on various Content recoginition machine learning techniques.

In this paper we have made following contributions:

- We have tried various types of noises on images and studied their effect on various classifier that can be used to break captcha
- We have done a study of SVM based Classifier[1] and also on other real world API's that are implemented using Convolutional Neural Network and their responses to noisy images

II. Motivation

Most of the Image Captcha's we have mentioned asks user to classify around 12 images correctly. If we have a classifier which can classify particular image with some accuracy, say x. Then the probability of guessing all the 12 images correctly is

$$f(x) = x^{12} \tag{1}$$

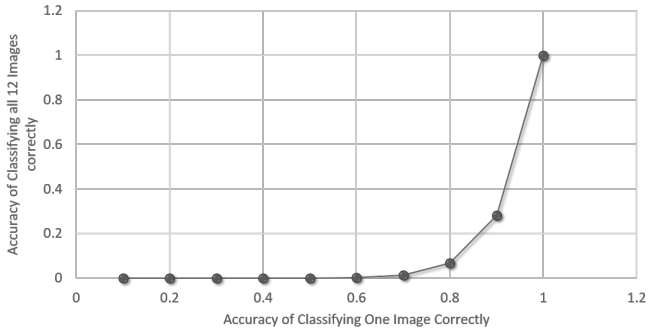

Fig. 2. This graph shows how the reduction in accuracy of classyfying one image can exponentially reduce the accuracy of classifying all the 12 images correctly.

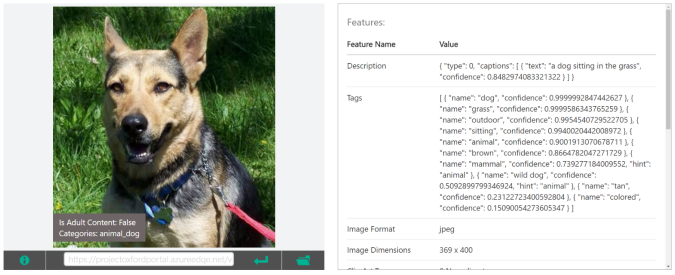

Fig. 3. This figure shows how Microsoft's Computer Vision API detects the content in an image with very high confidence

Hence in this paper we aimed to reduce the accuracy of different classifier with different types of noises. We then study the reduction in accuracy for these classifier. To achieve this goal we counter-attack these machine learning techniques. We reduce the classification accuracy by causing a concept drift. Concept drift is nothing but changing the statistical properties of targeted variable such that it causes the prediction made by these machine learning technique to be less accurate.

III. Attacks

We have roughly classified these attacks (or Machine Learning Classifiers) in two parts. In first part we study SVM based classifier which uses color features to classify images. While in second, we study we study attacks by Covolutional Neural Network which is more feature rich and has very high accuracy.

A. SVM Based

Wikipedia defines SVM (Support Vector Machine) as supervised learning models with associated learning algorithms that analyze data used for classification and regression analysis. Given a set of training examples, each marked for belonging to one of two categories, an SVM training algorithm builds a model that assigns new examples into one category or the other, making it a non-probabilistic binary linear classifier[7]. In 2008 Golle et al. [1] proposed a SVM specifically costumed for ASIRRA. The paper used color features and texture features from images for classification. In our evaluation we have used color feature part of this SVM. We have not tested it for texture features since it takes too long to extract texture features and build the matrices. Even the color features extraction and classification takes hours.

Now we shall see how the flow of this SVM [1] works. First they re-size all the images in 250-by-250 resolution.Then they add white borders to image to make it fit in square shape aspect ratio. After that they divide image in N horizontal and N vertical strips of equal width. Due to these strips we get N^2 cells. They have used HSV(hue, saturation,value) model where they divide hue channel into color spectrum C_h, C_s and C_v bands. Altogether this gives a partition of the whole color space into $C_\mathrm{h}C_\mathrm{s}C_\mathrm{v}$. Feature vector associated with an image indicates that for every cell in the image and every color region there is at least one pixel in cell which belongs to the color region. The feature vector $F(N,C_h,C_s,C_v)$ is a boolean vector for each cell. This SVM classifier has accuracy of 77.1%.

B. Convolutional Neural Network Based

Korekado, Keisuke, et al.[8] explain convolutional neural network for image recognition as consisting of multiple layers of small neuron collections which process portions of the input image, called receptive fields. The outputs of these collections are then tiled so that their input regions overlap, to obtain a better representation of the original image; this is repeated for every such layer. Tiling allows CNNs to tolerate translation of the input image. In this paper we tested various Image tagging API's , this includes Microsoft Cognitive Services[9], Clarifai[10], IBM's Alchemy API[11], Google's Reverse Image Search[12] and Imagga[13].

IV. Noise in Image Captcha

Motivation behind adding a noise to these image is human ability of imagination to fill the missing pieces in image to form object. For example, If you have a car drawn and you erase few spots of the drawing. The person watching that drawing can still make out that its a car. We add noise to images as shown in Figure 4. As we can see, despite having all these noises it is apparent that the animal in that picture is dog. We have tried introducing various types noises in images, we shall see those in details now. Figure 4 shows the effect of these various noises on a same image. Table 1, shows the accuracy for our implementation of SVM classifier for following types of noises.

1) Salt and Pepper Noise: In this case we added random noise to image. We chose random 5% pixels out of all pixels in image. Then we color those pixels with any random value out of 16 million colors from RGB-colorspace. As we have seen in SVM where algorithms checks whether one or more pixel in the cell falls in the color region. This will create false feature vector. When feature is supposed to take value 0 as there are no pixel from that region instead it takes value 1 for particular feature vector due to noisy pixel which in turn creates a Concept drift. We observed that in case of SVM, for six different feature set the drop in accuracy varied from 1.7% to 3.5% with average drop of 2.5%.

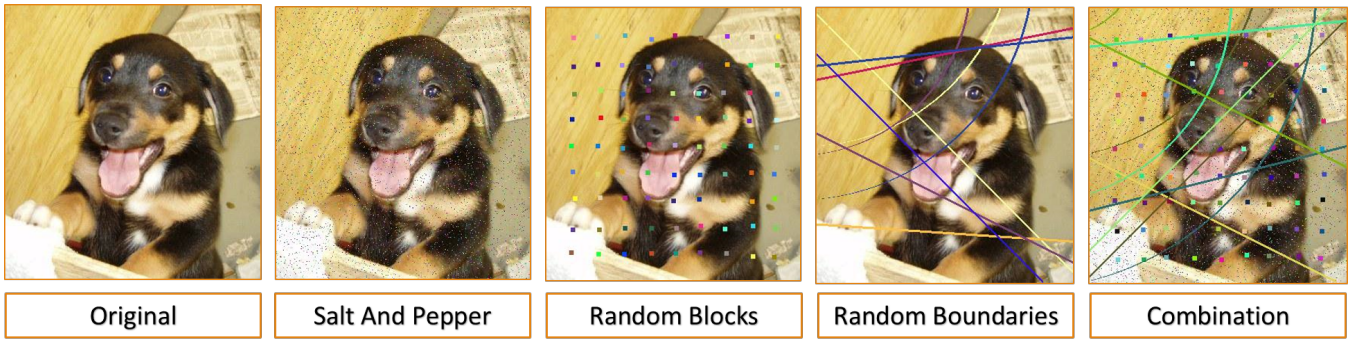

Fig. 4. Figure shows output image after applying various noise technique to image

2) Random Blocks: Here we divide image virtually in 10-by-10 blocks, creating 100 virtual boxes. Inside each box we put a block of random color at random location. The block's height and width we choose is 50th fraction of the original height and width of image. The block strategy was used to deceive color based feature extraction where they divide images in blocks similar to SVM classifier that we saw[1]. Since the feature selected are directly affected by dominating color in particular block this technique might replace the dominating color in few of those blocks. Resulting in wrong feature vector. Our evaluation shows that for six different feature set, drop in accuracy varies from 2.2% to 6.6% with average drop of 4.1% across all feature set.

3) Random Boundaries: In this case we add random lines and arcs. Many neural network image recognition system learn about objects in image automatically by learning about the boundary of the object. Since we are introducing random arcs and lines, it disrupts CNN feature extraction where it tries to learn about shape of object by detecting its boundaries. For the experimentation purposes we added 5 lines with thickness of 3 pixels. Also we draw 3 to 5 arcs in image with again a 3 pixel length and with random color. In case of our implementation of SVM classifier, the accuracy drop across different feature vector was between 4.7% to 7.9%.

4) Combination of all: Here we applied all the noise filters to image that we saw above. Adding salt and pepper noise, random blocks and also random boundaries. The resultant picture has noticeable amount of noise in image. As we can see in Figure 4, despite this noise we can still identify the animal in picture. We have got exceptional results for these images on CNN. While on SVM classifier the accuracy drop compared to original image was between 7% to 10.9%.

V. Evaluation

We have done our testing in two different setups. In first setup, we tested our noisy images on SVM classifier which is based on color feature given by Philippe et al.[1]. While in second setup we tested our image with different image tagging API's that we listed in Section 3 B.

In SVM based testing first we run the classifier on original data without introducing any noise in image. These results are listed in Table 1. For different values of N,C_h,C_s,C_v with different size of training dataset we calculate the classifier accuracy. For example, feature set $F_3 = F(5,10,6,6)$ consists of 9,000 color features obtain by dividing image into 25 blocks (F = 5) and we divide color space in 360 color regions (C_h = 10, C_s = 6, C_v = 6). We get accuracy of 74.2% with 5,000 images in training dataset and with 10,000 images in training dataset accuracy increases to 75.7%. We have also combined F_1 , F_2 , F_3 which now has 15,760 total color features. Accuracy for this $F_1 \cup F_2 \cup F_3$ is 75.7% for original images with 5,000 images in training set and 76.9% with 10,000 training images.

For testing the accuracy for our noisy image we train SVM with noisy images. That means we do whole process of SVM from scratch for each type of noisy images. The assumption here is attacker gets hold of captcha images by scrapping the login page. Hence he can train his classifier with noisy image that we are going to use in captcha. Reduction in accuracy can be seen in Table 1 for various types of noises.

We also tested accuracy of various image tagging API's . Note that since most of these API's are paid we were not able do thorough testing on these API's. For testing purposes we selected random 10 images which included 5 cat images and 5 dog images. we feed these images to these API's and see if it can classify the image in either cat or dog. We found that Microsoft Cognitive Service's Vision API is most accurate and most robust to various noises and it gave correct results despite the the various types of noises. Other classifier did fail to classify few images with noises. One of the notable conclusion from this study was that all the API's including Microsoft's Vision API failed miserably to classify images whihc all the types on noises combined. Clarifai and Imagga failed to classify even a single image correctly. Hence we can conclude that our technique is effective and classifier's which use more complex machine learning technique such as CNN also fail to break the captcha.

VI. Limitations

We can observe through various technique and testing that it is possible to reduce the accuracy of various classifiers. Our evaluation does show that our noise technique in image causes reduction in correct classification. But we would like to point out that our testing is not through, specially in case of classifier which uses more complex algorithms such as convolutional

TABLE I
SVM CLASSIFIER ACCURACY FOR DIFFERENT TYPES OF NOISES

Color features						# Images		Classifier accuracy				
Feature Set	N	C_h	C_s	C_v	# features	Total	Training Set	original	Salt Pepper	Blocks	Boundaries	Combination
F_1	1	10	10	10	1000	5,000	5,000	66.7%	63.8%	60.1%	58.8%	55.8%
F_2	3	10	8	8	5,760	5,000	5,000	75.1%	71.6%	71.0%	67.8%	66.2%
F_3	5	10	6	6	9,000	5,000	5,000	74.2%	71.9%	72.0%	69.5%	67.2%
F_3	5	10	6	6	9,000	10,000	10,000	75.7%	73.4%	73.2%	70.2%	68.5%
$F_1 \cup F_2 \cup F_3$					1000	5,000	5,000	75.7%	74.0%	70.8%	68.1%	67.5%
$F_1 \cup F_2 \cup F_3$					1000	10,000	10,000	76.9%	74.8%	72.7%	69.7%	69.5%

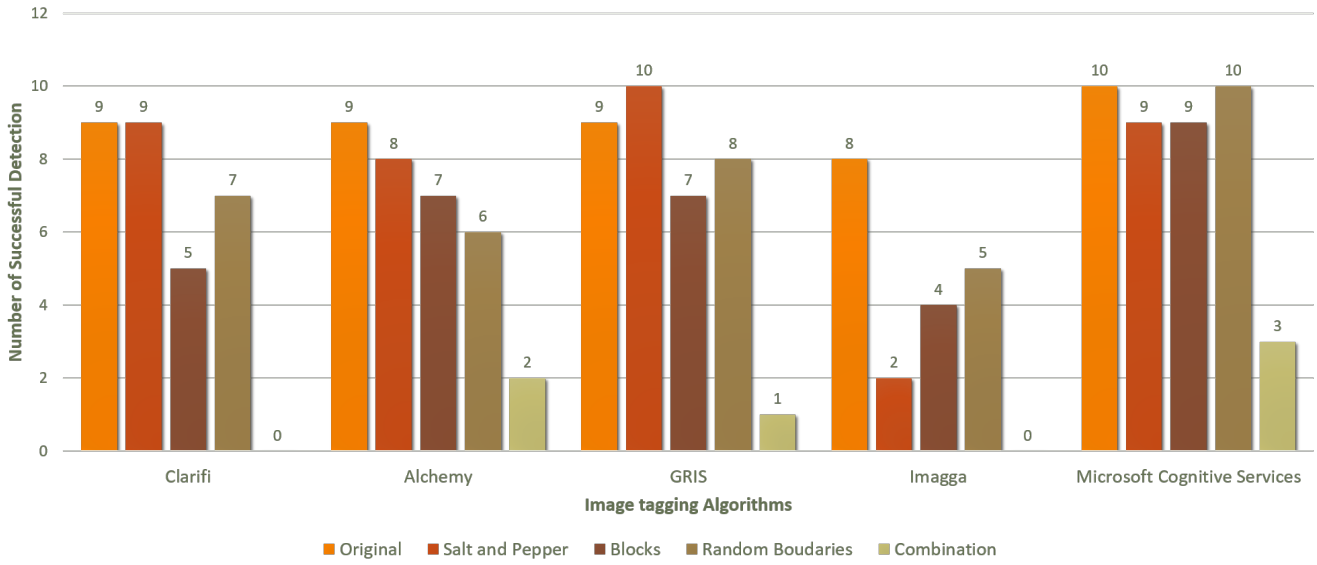

Fig. 5. This figure shows number of correct classification for cat or dog by various image tagging API's

neural network. We tested only 10 images in each of these API's due to the lack of access to these API. Also we would like to point out that unlike SVM where we trained SVM with noisy images, we did not do that for CNN. We propose a future work where we are planning to use caffe[15] with noisy images as training dataset. It will be interesting to see by how much does it reduce the accuracy in those cases. We would also like to point out that in September 2013 kaggle hosted a competition "Create an algorithm to distinguish dogs from cats" [16]. Most of the contestants reached very high accuracy we intent to explore the algorithms implemented by participants if they are available publicly.

Our Defenses or introducing noise in image can also be attacked. One of notable attack could be a noise removal attack where attacker can remove noises from image. We would like to point out that removing salt and pepper noise from image can be done and original can be restored [16]. But we think that image with combination of different noises wont be easy to restore due to randomness of our noise introduction technique and high number of noisy pixels in image.

Also due to computational heavy nature of our implementation of SVM or matter of fact for any image tagging CNN's we were not able to test variation to noise introduction technique. We would like to see how much does the accuracy gets affected if we increase the number of noisy pixels in images from 5% to 10% or more. Or In case or of Random blocks technique we would like to see how the size of each block affects the overall accuracy. In case of Random Boundaries we would like to add more random shape boundaries instead of just lines and arcs. We leave this as our future work.

VII. CONCLUSION

Our experimental results with SVM and different Image tagging API's shows that machines learning technique's can very well classify images like us human with decent accuracy. Our results have shown that our noise introduction techniques are effective and can bring the best possible accuracy of SVM classifier from 76.9% to 69.5%. To put things into prospective original system had probability of getting all 12 images right was 4.3% and with our technique it comes down to 1.3%. Our results with other API's were even better. Most of the API's were able to correctly classify almost all the images. But with our combination noise they fail miserably as only 2 or 3 images were classified correctly by API's and few of them failed to classify any of those noisy images. Hence we

will like to conclude that we were able to reduce the accuracy for the attacks on captchas.

ACKNOWLEDGMENT

I would like to thank Prof. Roberto Perdisci for suggesting us the topic and giving guidance for this project. I would also like to thank my classmates for their valuable feedback during class presentations.

REFERENCES

[1] Golle, Philippe. "Machine learning attacks against the Asirra CAPTCHA." Proceedings of the 15th ACM conference on Computer and communications security. ACM, 2008.
[2] https://en.wikipedia.org/wiki/CAPTCHA
[3] http://research.microsoft.com/en-us/um/redmond/projects/asirra/
[4] Elson, Jeremy, et al. "Asirra: a CAPTCHA that exploits interest-aligned manual image categorization." ACM Conference on Computer and Communications Security. 2007.
[5] http://research.microsoft.com/en-us/um/redmond/projects/asirra/installation.aspx
[6] https://www.google.com/recaptcha/intro/index.html
[7] https://en.wikipedia.org/wiki/Support_vector_machine
[8] Korekado, Keisuke, et al. "A convolutional neural network VLSI for image recognition using merged/mixed analog-digital architecture." Knowledge-Based Intelligent Information and Engineering Systems. Springer Berlin Heidelberg, 2003.
[9] https://www.microsoft.com/cognitive-services/en-us/computer-vision-api
[10] https://www.clarifai.com/
[11] http://www.alchemyapi.com/
[12] https://support.google.com/websearch/answer/1325808?hl=en
[13] https://imagga.com/
[14] Sivakorn, Suphannee, Jason Polakis, and Angelos D. Keromytis. "Im not a human: Breaking the Google reCAPTCHA."
[15] Jia, Yangqing, et al. "Caffe: Convolutional architecture for fast feature embedding." Proceedings of the ACM International Conference on Multimedia. ACM, 2014.
[16] Chan, Raymond H., Chung-Wa Ho, and Mila Nikolova. "Salt-and-pepper noise removal by median-type noise detectors and detail-preserving regularization." Image Processing, IEEE Transactions on 14.10 (2005): 1479-1485.
[17] https://www.kaggle.com/c/dogs-vs-cats