Adversarial Attacks on Image Captchas

Jessal V. A., Adhithya S., Janaki Keerthi, Jayasoorya Jithendra

Department of Computer Science and Engineering , College of Engineering Trivandrum

An adversarial attack consists of subtly modifying an original image in such a way that the changes are almost undetectable to the human eye. The modified image is called an adversarial image, and when submitted to a classifier is miss classified, while the original one is correctly classified. The real-life applications of such attacks can be very serious –for instance, one could modify a traffic sign to be misinterpreted by an autonomous vehicle, and cause an accident. Previous adversarial examples have been designed to degrade performance of models or cause machine learning models to produce specific outputs chosen ahead of time by the attacker. We aim to introduce adversarial attacks that instead reprogram the target model to perform a task chosen by the attacker.Most successful attacks are gradient-based methods. Namely the attackers modify the image in the direction of the gradient of the loss function with respect to the input image. There are two major approaches to perform such attacks: one-shot attacks, in which the attacker takes a single step in the direction of the gradient, and iterative attacks where instead of a single step, several steps are taken. We aim to adopt the Fast Gradient Sign Method (FGSM) which computes an adversarial image by adding a pixel-wide perturbation of magnitude in the direction of the gradient. This perturbation is computed with a single step, thus is very efficient in terms of computation time.

Adversarial Attack | CAPTCHA | Deep Learning | Convolutional Neural Network | Fast Gradient Sign Method | Transfer Learning | MobileNet

INTRODUCTION

In the recent times, excellent results have been achieved in different real-world applications like autonomous cars, face recognition and medical image analysis, to name a few. These breakthroughs are not only due to the advances in Deep Neural Networks (DNNs), but also the availability of huge amount of data and computational power. Autonomous vehicles have become so reliable that they no longer need human drivers inside as a backup. Medical systems are now better than human experts in detecting cancer metastages. Even facial recognition software is able to surpass human capabilities. In spite of all these impressive advancements, the research community has recently found that Deep Neural Nets are susceptible to adversarial attacks.

Here we present our project on adversarial attacks using Deep learning. Deep Neural Networks are susceptible to adversarial noise. The motivation behind the project is to make use of this vulnerability in a constructive manner in order to improve computer security. Adversarial examples for captcha applications are very difficult for Deep Learning algorithms while easy for humans (the adversarial noise tends to be very small and does not affect human perception of image content). Machine learning models misclassify examples that have only a slight difference from correctly classified examples that are drawn from the data distribution. As a matter of fact in several cases, a wide range of models with different architectures trained on different subsets of the training data were found to misclassify the same adversarial example.

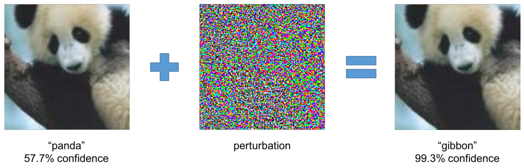

Fig. 1. Adversarial attack on the image of a panda

RELATED WORKS

Basically, CAPTCHAS are automatically constructed problems which are quite difficult to be solved by AI algorithms, but easy for humans. However, large number of CAPTCHA designs have become ineffective because of the fast growth in AI. Particularly, recent advancements in the field of deep learning reduced the gap between machine and human ability to solve problems that were usually used in CAPTCHAS in the past. As a matter of fact, several breakthroughs in AI led some researchers to claim that deep learning would gradually lead to the "end" of CAPTCHAS (1).

Despite all this, DNNs still have some shortcomings with regard to human capability (2). To be specific, they are susceptible to small perturbations in the input, that cannot be perceived by humans but still cause misclassification. Such perturbations, can be specifically crafted for a particular input that forces misclassification by the machine learning model. Though initially this phenomenon was discovered in the context of DNNs, it was observed in other classifiers also, such as decision trees, KNN etc.

Moreover, it was proved that adversarial examples designed to be misclassified by one model are often also misclassified by different other machine learning models (3). This transferability allows adversarial examples to be used in misclassification attacks on machine learning systems, even without having access to the underlying model (4). Consequently, adversarial attacks pose serious security threats for a number of machine learning based solutions like facial recognition, biometric authentication, voice commands and spam filters. However, this project aims to make use of adversarial examples in a constructive way so as to improve computer security.

ADVERSARIAL ATTACKS

Modern neural networks manipulate data in the form of 32-bit floating point numbers. Whereas modern hardware manipulate images as 8 bit values. Thus adversarial attacks manipulate the data such that the main 8 bits are unchanged, by modifying the remaining 24 bits. Neural Networks perform linear transformation to the matrices. Additions of pre-training, dropuots, and model averaging do not improve the vulnerability of model to adversarial examples. Convolutional neural networks approximate Perceptual distance as Euclidean distance. This resemblance has a clear flaw if images that have an immeasurably small perceptual distance correspond to completely different classes in the network's representation. The linear examples of adversarial attacks are yet to be discussed. The features are of limited precision. It is irrational for the classifier to respond differently to an input x than to an adversarial input $x \quad = x + \eta$ if every element of the perturbation η is smaller than the precision of the features. Precision of the features is about (1/256) ≈ (0.03). Thus for $\eta \leq (0.03)$ the adversarial input should act similiar to the original input. Thus effectively the transformation becomes:

$$W^T * \widetilde{x} = W^T * x + W^T * \eta$$

PROPOSED SYSTEM

Phases. The main phases of the system are as follows:

- **CAPTCHA Generation** A CAPTCHA generator is used to give an input CAPTCHA to the system. The CAPTCHA consists of 6 characters in text format.

- **Image Modification** This is where adversarial noise is added to the image using FGSM. The modification will be imperciptible to human eye.

- **CNN Model prediction** Modified image is given to a Convolutional Neural Network model and prediction is made. Predicted value is compared with the original value and performance of system is evaluated.

- **Output** Output is a modified CAPTCHA image, which is indistinguishable to human eye, but wrongly predicted by the CNN model.

A python script was used to generate the CAPTCHAs. The script can be used to generate CAPTCHAs consisting of lowercase alphabets or uppercase alphabets or digits. It also allows us to generate CAPTCHAs of different lengths. For this project, the CAPTCHA length was restricted to 6 lowercase characters. The generated CAPTCHAs were then subjected to modification using the FGSM and given to the CNN model.

.

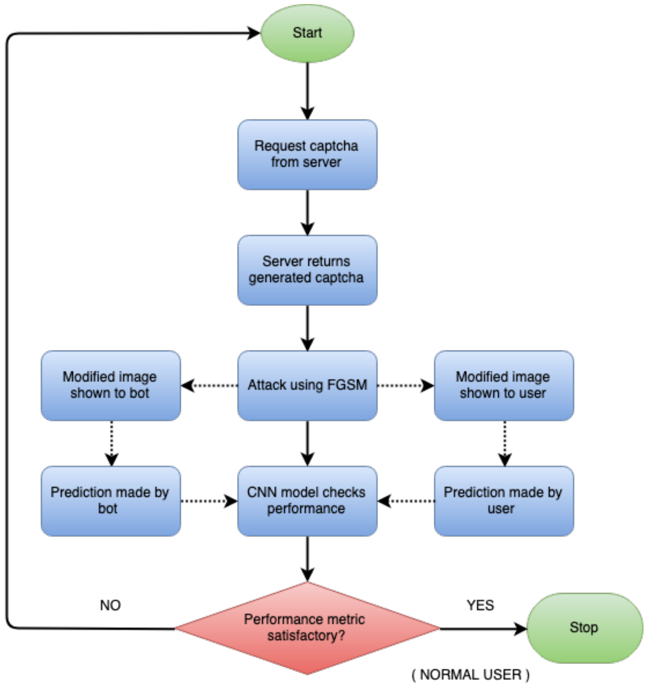
Start
Request captcha from server
Server returns generated captcha
Modified image shown to bot
Attack using FGSM
Modified image shown to user
Prediction made by bot
CNN model checks performance
Prediction made by user
NO
YES
Performance metric satisfactory?
Stop
(NORMAL USER)

Fig. 2. Basic Model

The Fast Gradient Sign Method. FGSM computes an adversarial image by adding a pixel-wide perturbation of magnitude in the direction of the gradient. This perturbation is computed with a single step, thus is very efficient in terms of computation time.

$$X^{adv} = x + \epsilon.sign(\nabla_x J(x, y_{true}))$$

where,
X : is the clean input
X^{adv} : is the perturbed adversarial example.
J : is the classifier's loss function.
y_{true} : is the true label for the input x.

Targeted Fast Gradient Sign Method is similar to the FGSM, in this method a gradient step is computed, but in this case in the direction of the negative gradient with respect to the target class:

$$X^{adv} = x - \epsilon.sign(\nabla_x J(x, y_{target}))$$

where y_{target} is the target label for the adversarial attack.
In the Iterative Fast Gradient Sign Method, The iterative methods take T gradient steps of magnitude $\alpha = \epsilon/T$ instead of a single step t:

$$X_0^{adv} = X$$

$$X_{t+1}^{adv} = X_t^{adv} + \alpha.sign(\nabla_x J(x, y_{true}))$$

Transfer learning and the MobileNet model. Transfer learning is a popular deep learning method wherein pre-trained models are used as the starting point. Instead of training a deep network from scratch for the task, it allows to take a network trained on a different domain for a different source task and adapt it for the target domain and target task.

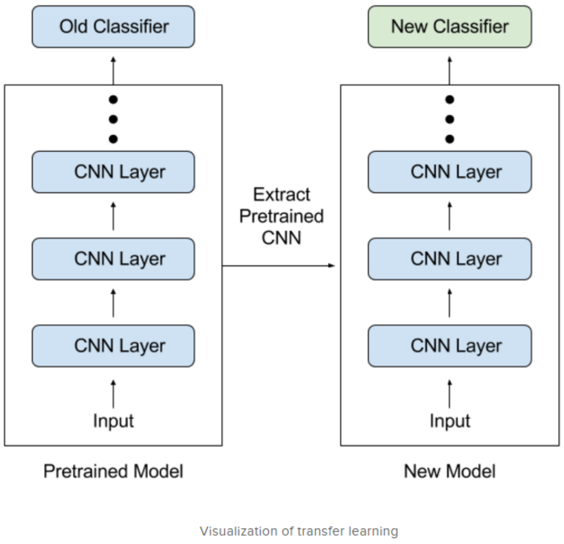

Fig. 3. Transfer Learning

For this, the MobileNet V-2 architecture was used. The core layer of MobileNet consists of depthwise separable filters called the Depthwise Separable Convolutions. The network structure is another factor that boosts the performance. Also the resolution and width can be tuned to trade off between accuracy and latency.

MobileNet-V2 is capable of improving the performance of mobile models on multiple tasks and also across a wide range of various model sizes. Depthwise separable convolutions which are basicallya form of factorized convolutions, factorize a standard convolution into a depthwise convolution and a 1×1 convolution called a pointwise convolution. In MobileNet, the depthwise convolution applies a single filter to each input channel. A 1×1 convolution is then applied by the pointwise convolution so as to combine the outputs of the depthwise convolution.

MobileNet-V2 utilizes a module architecture similar to the residual unit with bottleneck architecture of ResNet; the modified version of the residual unit where convolution3×3 is replaced by depthwise convolution. Contrary to the standard bottleneck architecture, the first convolution 1×1 increases the channel dimension, then depthwise convolution is performed, and finally the last convolution 1×1 decreases the channel dimension

Training performed and Results obtained. Initially 10gb dataset was uploaded on FloydHub and training was performed. It was successful but resultant accuracy was only about 25% . Inorder to improve accuracy, we decided to use a pretrained model (Transfer Learning). Training was then performed on Google Colaboratory by making use of the MobileNet V2 model. In Google Colab, the dataset was compressed using tfrecord where 10gb data can be compressed to approximately 1gb training tfrecord. The training performed on Google Colab was successful and reached upto 99% accuracy for each of the 6 characters.

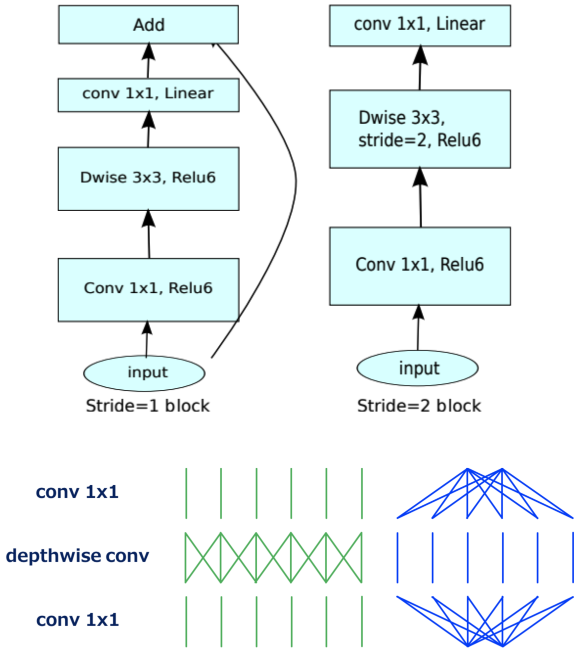

Fig. 4. MobileNet V2 architecture

Since the loss is decreasing and the accuracy is found to be increasing, it indicates that the model build is learning and working fine. The accuracy we got during the training of the dataset is given in the table.

Epoch	dense_loss	dense_acc	val_acc
1	0.1701	0.9511	0.0526
2	0.0035	0.9993	0.9998
3	0.0020	0.9996	0.9984
4	0.0013	0.9997	0.8964
5	7.9859e-04	0.9998	0.9826

Table 1. Accuracy status after training

DISCUSSION AND FUTURE PERSPECTIVES

In this paper, we proposed an effective alternative defence scheme against captcha solving bots and thereby reducing the threats. Because, many of the bots which are capable to solve captchas are growing exponentially, it prompts the real need of having a comprehensive solution.

In the world of existing advanced AI, there exists a security threat to CAPTCHAs as mentioned. This system implemented in the form of a web application, provides a simple and secure mechanism to the existing threat by deceiving the deep learning tools. The same model can be used to secure social media pictures from being automatically tagged by bots. Additionally, adversarial networks can have a wide range of applications and is sure to lead to new research outcomes.

SOFTWARE AND HARDWARE AVAILABILITY

All the files regarding the system has been uploaded and the source is given below. Each of its modules can be installed by enabling the corresponding code repository or by following the instructions on the corresponding website:

- https://github.com/jessalva/Adversarial-Attacks-Final-Year-Project

ACKNOWLEDGEMENTS

Firstly, we would like to thank the Almighty for giving us the wisdom and grace for presenting our project design.

With a profound sense of gratitude, we would like to express our heartfelt thanks to our guide **Prof. Rajasree R.**, Assistant Professor, Department of Computer Science and Engineering for her expert guidance, co-operation and immense encouragement in pursuing this project presentation.

We are very much thankful to **Dr. Salim A.**, Head of Department and **Prof. Vipin Vasu A. V.**, Assistant Professor of Department of Computer Science and Engineering, our guide for providing necessary facilities and his sincere co-operation.

Our sincere thanks is extended to all the teachers of the Department of CSE and to all our friends for their help and support.

BIBLIOGRAPHY

1. Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *Conference paper at ICLR*, March 2015.
2. Shixiang Gu and Luca Rigazio. Towards deep neural network architectures robust to adversarial examples. *arXiv preprint arXiv:1412.5068*, 2014.
3. Florian Tramèr, Alexey Kurakin, Nicolas Papernot, Ian Goodfellow, Dan Boneh, and Patrick McDaniel. Ensemble adversarial training: Attacks and defenses. *Conference paper at ICLR*, July 2018.
4. Gamaleldin F. Elsayed, Ian Goodfellow, and Jascha Sohl-Dickstein. Adversarial reprogramming of neural networks. *CoRR abs/1806.11146*, November 2018.