Adversarial Path in Browser based classification

Koushik Chowdhury (2572865)
Machine Learning Cybersecurity Project Showcasing 2021
Saarland University, Germany

[1]s8kochow@stud.uni-saarland.de

Abstract. *This report presents a web-based classification in the browser using the MobileNet model. The mobileNet model is proposed by Google. This architecture has low latency and low power and is used popularly for classification, segmentation, and detection. It works almost the same as traditional convolution but is much faster. And as for dataset, we have used ImageNet, which is a large visual database. This database is basically used for object recognition tasks and research. For the next adversarial part, we used a universal adversarial patch. The patch can be used to attack any classification or image, and this is why it is named the so-called universal patch. In the later part of our project, we introduced images with modified pixels and orientations to test the universal patch and its efficiency in attack.*

1. Introduction

Adversarial patch attacks are recently recognized as the most practical threat model against the real-world computer vision system. The vulnerability of deep learning systems to adversarial examples causes to change the output of a classification, which is mostly not visible to any human eyes. For our project, we have experimented with different patch shapes under different lights and angles and observed the robustness. It has been shown in a study that even when the adversarial patch is printed out, it still remains adversarial to image classification, as proved by Kurakin et al. [1]. For defense, some study [2][3][4] has also been conducted to increase the adversarial robustness of image models to small Lp perturbations to the input. In the beginning of this project, we created a browser-based classification model. The sole purpose behind browser-based classification was that users with no coding capabilities could use our project and test the hypothesis using their data. JavaScript was used to convert the machine learning model into a browser and later tested with random images where an adversarial patch was introduced.

2. Background Study

This research went through some of the state-of-the-art research papers. One of the research papers has shown ML systems are vulnerable to adversarial examples, and in their research, they fed images (adversarial) from a mobile phone camera to an ImageNet classification, and at the end, they measured the classification accuracy of the system [1]. Some MIT researchers studied the adversarial robustness of deep learning [2]. They trained networks on two different datasets that are robust to a wide range of adversarial attacks. Also, this research went through a publication from Google based on adversarial patches [5], where they worked with a universal patch and tested it against random images. From these papers, we have gathered information about adversarial patch attacks and defenses. Furthermore, we also had to wrap our heads around the concept of browser-based classification and how to focus on the presentation of a server-side machine learning model into a JavaScript client-side presentation.

3. Dataset

One of the preliminary parts of this work was data collection. The ImageNet dataset was used, which has around 14 million images. Apart from that, another cool feature about the dataset is that we also have the option to add real data from random users since webcam features have been enabled in this project, which allows the user to tinker with the project and test any random dataset.

4. Approach

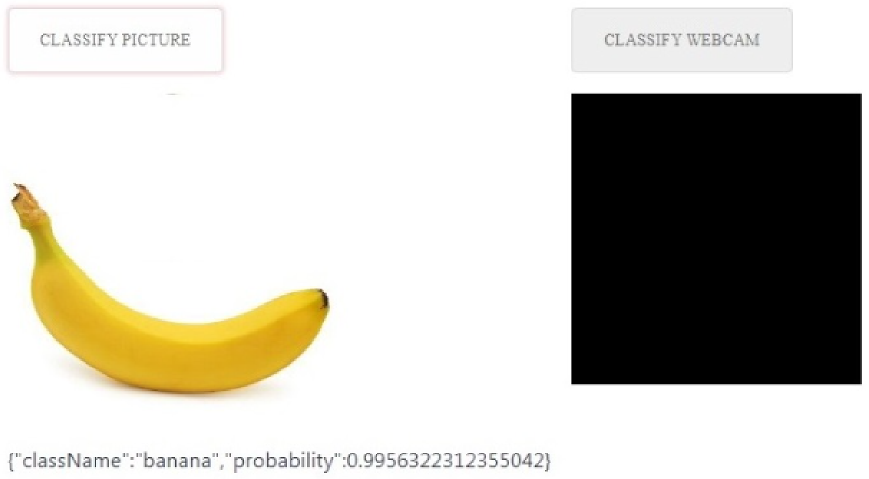
Since the project is browser-based, the Python model had to be converted to become compatible with browsers.Since the project is browser-based, the Python model had to be converted to become compatible with browsers. There were multiple options to choose from: the Python framework Flask or a state-of-the-art library called Tensorflow.js. This work considered the Tensorflow.js. The MobileNet model, which is the backbone of the classification model, is tagged in the script as well as the Tensorflow.js library. The whole project is divided into three parts: interface, back-end, and testing or attack.

For the interface part, the same old javascript is used, and HTML produces client-side output. At first, the model was loaded and made predictions through the model on the input images. An object from the Tensorflow.js data API was created that could capture images using the user's webcam as a tensor or any other camera for that matter. Once the image is captured, the tensor is disposed to release the memory. A kNN classifier was set up to handle the ability of the users so that they can add classes according to their will and their real-life dataset. To get this, the intermediate activation of MobileNet's 'conv_preds' is obtained and passed to the KNN classifier. Again, the tensor was disposed to release the memory and get the activation from MobileNet from the webcam. Thus the most likely class and confidence were gotten from the classifier module once tested. Then came the adversarial patch attack, for which we followed the traditional approach of finding a targeted adversarial example, given some input, a classifier, and a target class along with maximum perturbation. The attack was run by replacing a part of the classifying image with a universal adversarial patch. The patch was introduced to the image under different scaling, rotation, and lights. The classification with pre-trained adversarial patch images was also tested just to check if there were any differences in the results of the prediction.

5. Result

The evaluation baseline was defense using preprocessing input images that can be broken by white box adversaries. The success criteria were successful implementation of a browser-based classification model and testing of adversarial patch attacks. For the result part, the patch was used and rescaled to be placed at random locations on random ImageNet images. In the result, contradictory claims were found to most prior works as they claim the patch works for any background, which was not the case in our test; that goes against the universality of the patch. And as for the browser-based classification, the project runs on a local server and serves its purpose along with the webcam feature.

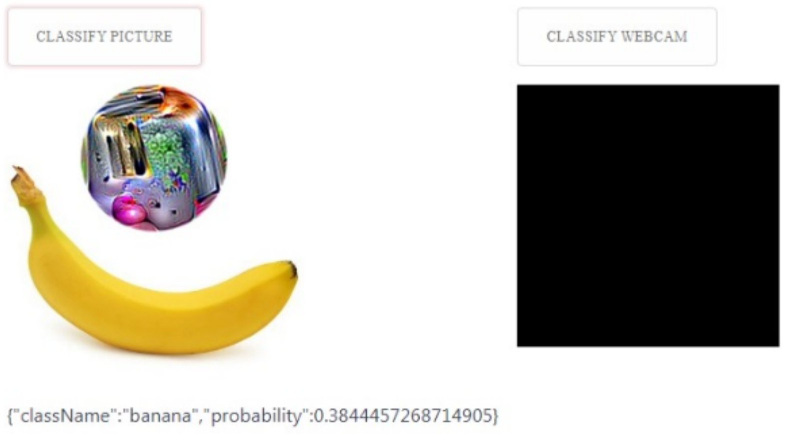
In the following example below, a normal picture of banana returns Mobilenet prediction accuracy at 0.99, which is a very high accuracy.



```
{"className":"banana","probability":0.9956322312355042}
```

Figura 1. Mobilenet prediction accuracy at 0.99

But in the next picture below, it is the same banana but with an added adversarial patch. Now this prediction accuracy drops drastically to 0.38, but it still predicts the outcome as banana, which was only possible after trying to reach the defense for adversarial attack to some extent.



```
{"className":"banana","probability":0.3844457268714905}
```

Figura 2. Prediction accuracy drops drastically to 0.38

6. Conclusion

In this project, the adversarial patch attack was demonstrated and shown that this patch can be universally threatening despite being of different shapes and sizes, random location, or random scaling, with some exceptions regarding different backgrounds. This work also focuses strongly on making the model public, and by public, I truly mean public, as this model can be tested and modified by any given user with or without any knowledge or understandability of machine learning. The user can truly implement their own data or images into the model, use adversarial patches on their classes to see the result, and be a part of the whole process.

Referências

[1] Kurakin A, Goodfellow IJ, Bengio S. Adversarial examples in the physical world. InArtificial intelligence safety and security 2018 Jul 27 (pp. 99-112). Chapman and Hall/-CRC.

[2] Madry A. Towards deep learning models resistant to adversarial attacks. arXiv preprint arXiv:1706.06083. 2017.

[3] Buckman J, Roy A, Raffel C, Goodfellow I. Thermometer encoding: One hot way to resist adversarial examples. InInternational conference on learning representations 2018 Feb 15.

[4] Tramèr F, Kurakin A, Papernot N, Goodfellow I, Boneh D, McDaniel P. Ensemble adversarial training: Attacks and defenses. arXiv preprint arXiv:1705.07204. 2017 May 19.

[5] Brown TB, Mané D. Aurko Roy, Martín Abadi, and Justin Gilmer. Adversarial patch. In31st Conference on Neural Information Processing Systems (NIPS 2017) 2017.