Fooling AI image classifiers

Outmani Ossama
Dept. of Mathematics & Computer Science, ENSAH College
Abdelmalek Essaadi Univ.
Al Hoceima, Morocco
ossama.outmani@etu.uae.ac.ma

Supervised by Prof. Khamjane Aziz
Dept. of Mathematics & Computer Science, ENSAH College
Abdelmalek Essaadi Univ.
Al Hoceima, Morocco
akhamjane@uae.ac.ma

Abstract— Often , when building an AI image classifier ,we tend to evaluate the resulted model by measuring its accuracy on the test-set . However is this approach enough ? In this paper we tested the ability of MobileNetV2 to correctly classify carefully generated adversarial examples to harm the model .We observed it's weakness in all tests we did .Which leads as to the need to evaluate this kind of AI models taking in consideration this type of input images.

Keywords—computer vision, adversarial examples, image classification

I. INTRODUCTION

As artificial intelligence (AI) image classifiers become increasingly prevalent, their susceptibility to adversarial examples emerges as a critical concern. This study narrowly focuses on the Fast Gradient Sign Method (FGSM) attack, a potent technique that subtly alters input images to induce misclassifications. The FGSM attack, originally introduced by this paper [1], stands as a representative threat to the robustness of image classifiers. Within this context, our research aims to rigorously test and analyze the impact of FGSM attacks on AI image classifiers. While our focus is on this specific attack, which is a white box attack requiring have access to model parameters, there exists more types like black box attack ,targeted attacks and untargeted attacks ...

II. EXPERIMENTS

A. *The targeted model*

In the following experiences ,we will be testing the robustness of MobileNetV2 model against adversarial examples .MobileNetV2 is a CNN that is 53 layers deep trained on the ImageNet Dataset with more than one million labeled images .It can classifies input 224x224 images into 1000 classes .

B. *Experiments*

The code of all the Experiments mentioned in this paper are openly available throught Google Collab [2].All the assets used including the source image are available on Github [3]

1) *Experiment 1:*

In the pursuit of testing the robustness of the model, we subjected it to an FGSM (Fast Gradient Sign Method) attack . This attack involves the introduction of perturbations to the input image using the following formula:

$$f(x) = x + \epsilon sign(\nabla_x J(\theta, x, y))$$

where :

- x is the matrix representing the image under consideration.
- ϵ multiplication factor used to control the amount of noise applied to the original image
- f symbolizes the function responsible for generating the adversarial example.
- θ the parameters of a model
- y represents the targets associated with x.
- J(θ,x,y) denotes the loss function, specifically the Categorical Cross Entropy loss in our scenario.

Result 1.1 :

As shown in "Fig. 1." ,the attack changed the classification label ,which means that the model despite his efficiency and the large input of data used in training was susceptible to the introduced attack.

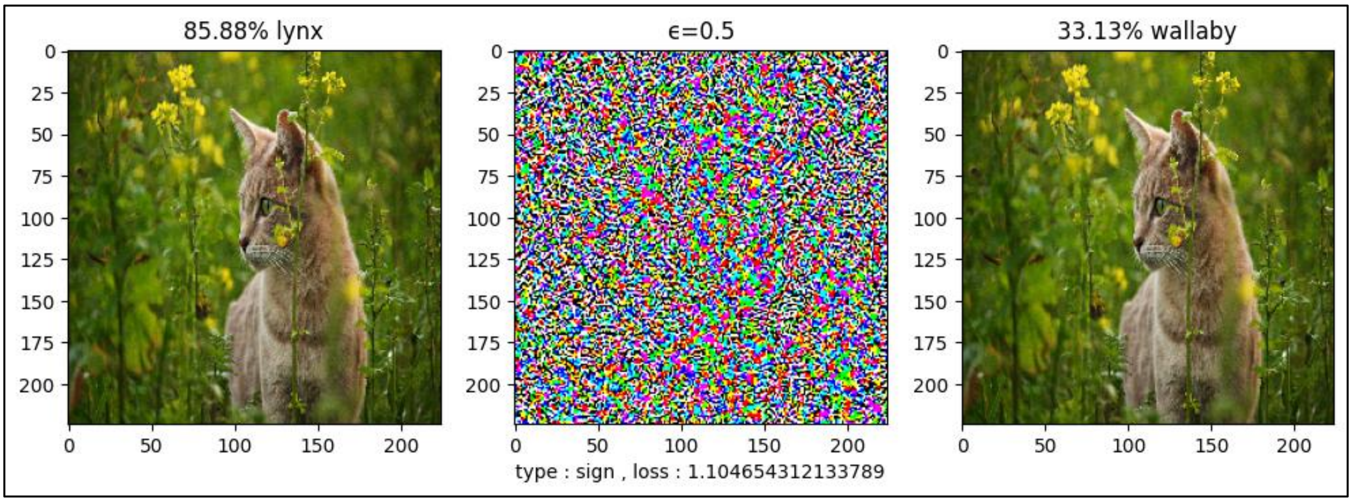

Fig. 1. **Adversarial example generated with ϵ=0.5 ,where "type=sign" stands for using the standard FGSM technique**

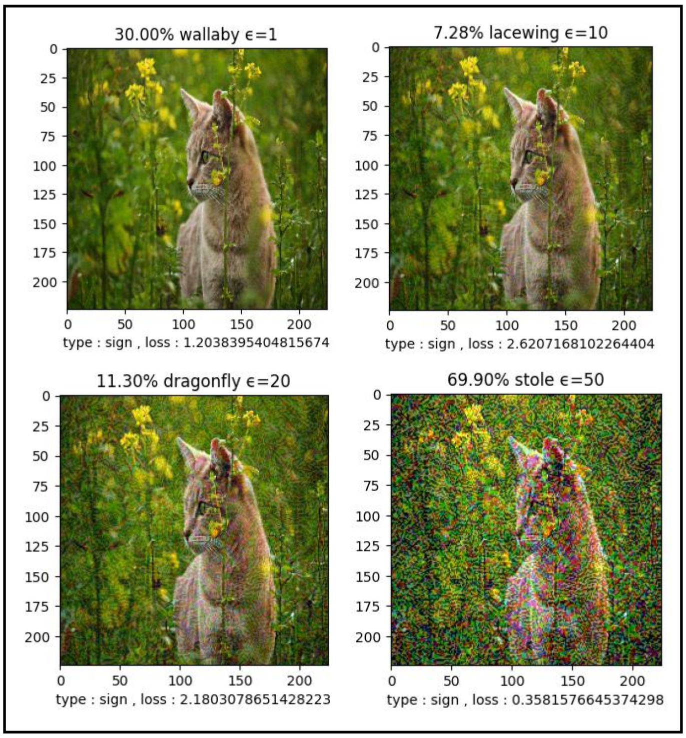
30.00% wallaby ϵ=1
type : sign , loss : 1.2038395404815674
7.28% lacewing ϵ=10
type : sign , loss : 2.6207168102264404
11.30% dragonfly ϵ=20
type : sign , loss : 2.1803078651428223
69.90% stole ϵ=50
type : sign , loss : 0.3581576645374298
0
25
50
75
100
125
150
175
200

Fig. 2. **Adversarial examples generated with multiple values of ϵ ,and their classification**

2) *Experiment 2:*

In order to the test further model's resilience, we decided to find a better value of ϵ ,so we did tests on multiple values ,the result is shown in "Fig. 2." .

Result 2.1 :

Larger values of the perturbation factor ϵ can lead to a higher loss ,however we get more noticeable noise , so there is a trade-off between the amount of distortion and the harm we want to cause to model.

Result 2.2 :

Even if ϵ is getting higher ,it was observed that the loss metric could decrease. We can justify this behavior with the non-convex nature of the loss variation concerning at least some pixels.To fix this problem, a proposed solution could be the Projected Gradient Descent (PGD) technique, which is an iterative version of FGSM .We choose a small value of ϵ, then we apply perturbation on multiple small steps in the direction of the steepest ascent, hence we will not miss the local maximum

3) *Experiment 3:*

In our pursuit of addressing the trade-off highlighted in Result 2.1, we aimed to optimize the FGSM technique by introducing a variable adjustment to pixels based on their influence on the loss. To implement this, we proposed the utilization of a normalized gradient format. This approach was designed to minimize distortion on pixels with lesser impact on the prediction, thereby achieving a more nuanced and targeted perturbation. The modified formula is expressed as follows:

$$f(x) = x + \epsilon(\nabla_x J(\theta, x, y)/\max_i(|x_i|))$$

where :

- xi represents each element of the x matrix of the given image

Result 3.1:

With same value of ϵ ,we have very less impact on image as shown in "Fig. 3." , so we can now apply bigger values of ϵ without too much affecting the image

Result 3.2:

With same values of ϵ compared to the standard FGSM ,we are getting less value of loss .Which is logical since now the less important pixels are less affected with noise .To solve this problem , we can simply use bigger values of ϵ without too much affecting the image thanks to the normalization as shown in "Fig. 4.".

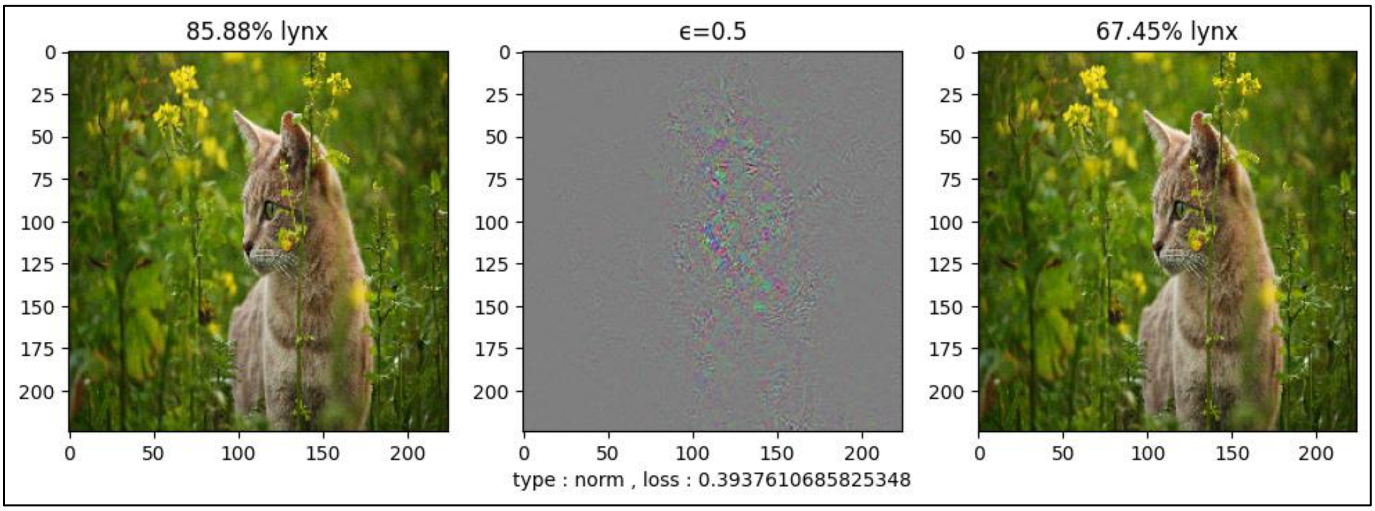

Fig. 3. **Adversarial example generated with ϵ=0.5 ,where "type=norm" stands for using the new normalized version of FGSM**

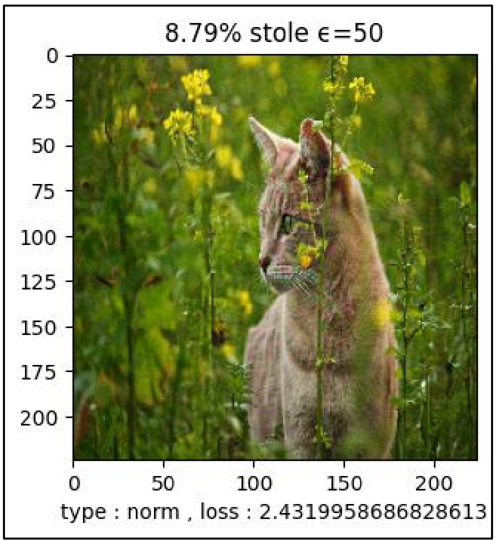

Fig. 4. **Adversarial examples generated with a high value of ϵ=50 , without too much noticeable noise as before**

Experiment 4:

To get a final idea about the level of harm an adversarial example can cause to the model , the original image and adversarial examples were printed out , then photographed with standard mobile phone and passed to the model to re-predict the class of each image as shown in "Fig. 5.".

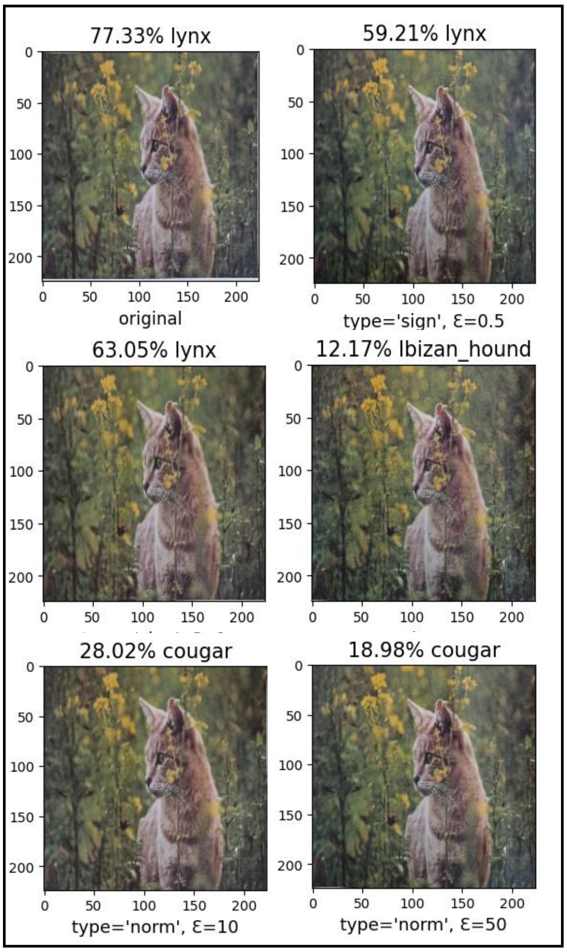

Fig. 5. **Multiple tests of the model's prediction on the original image ,standard and normalized FGSM versions with multiple values of ϵ, printed and photographed**

Result 4.1:

The model is still predicting the image as "lynx" , same as before , which is a good sign of the performance of the model ,when dealing with non adversarial examples.

Result 4.2:

the model is correctly classifying the adversarial exemples with lower amount of perturbation ,which is justifiable due to the big effect of printing and photographing on the colors of the images.

Result 4.3:

As soon as we increase ,the amount of noise by controlling the value of ϵ for either FGSM or the introduced Normalized version of it, the model is again misclassifiying the adversarial examples even if they are affected with printing and photographing quality ,which indicates how weak it is against adversarial examples .

III. Conclusion

MobileNet V2 despite it's power on classifying images due to the large amount of data he was trained on , failed in all the experiments we did to test its robustness .We can conclude that more work should be done after building Ai image classifiers ,in term of securing them against attacks with adversarial examples , especially if deployed in production.

IV. Acknowledgment

I sincerely thank our Professor Mr Khamjane Aziz for giving us this opportunity to choose freely any topic as subject of the project for his module "Machine learning" , and guiding us towards searching in research papers .It was a very helpful experience , discovering the world of scientific research in the world of artificial intelligence.

V. References

[1] I. J. Goodfellow, J. Shlens and C. Szegedy "Explaining and Harnessing Adversarial Examples"

[2] O. Outmani *Fooling AI image classifiers* [Google Colab].Available: https://drive.google.com/file/d/1QSzbfJLUHtCPWd3Ir7OeWSwigERWsC_G/view?usp=sharing

[3] O. Outmani *Adversarial-Exemples-in-Computer-Vision* [Github]. Available: https://github.com/nexossama/Adversarial-Exemples-in-Computer-Vision