## Lab 34: Using CleverHans with FGSM Attack

**Using CleverHans with FGSM Attack**

**Lab Manual**



**Disclaimer: The content is curated from online/offline resources and used for educational purpose only**

***Steps to implement CleverHans Library***

1. Visit the link: <https://colab.google/>
2. Click on ‘New Notebook’
3. Start typing the code given below
4. ***Installing the libraries***

pip install cleverhans

pip install tensorflow

1. ***Implementing the code for attack and visualization***

import numpy as np

import tensorflow as tf

import matplotlib.pyplot as plt

from tensorflow.keras.applications import MobileNetV2

from tensorflow.keras.applications.mobilenet\_v2 import preprocess\_input, decode\_predictions

from tensorflow.keras.preprocessing import image

from cleverhans.tf2.attacks.fast\_gradient\_method import fast\_gradient\_method

# 1. Load pre-trained MobileNetV2 model

model = MobileNetV2(weights="imagenet")

# 2. Load image from Colab upload

img\_path = "cat.jpg"  # Make sure this matches the name of the uploaded file

img = image.load\_img(img\_path, target\_size=(224, 224))

x = image.img\_to\_array(img)

x = np.expand\_dims(x, axis=0)

x\_preprocessed = preprocess\_input(x)

# 3. Get original prediction

preds = model.predict(x\_preprocessed)

original\_label = decode\_predictions(preds, top=1)[0][0][1]

# 4. Create adversarial example using FGSM

epsilon = 0.05

adv\_x = fast\_gradient\_method(model\_fn=model, x=x\_preprocessed, eps=epsilon, norm=np.inf)

# 5. Get adversarial prediction

adv\_preds = model.predict(adv\_x)

adv\_label = decode\_predictions(adv\_preds, top=1)[0][0][1]

# 6. Plot original and adversarial images

plt.figure(figsize=(10, 5))

plt.suptitle("Original vs. Adversarial Image", fontsize=16)

plt.subplot(1, 2, 1)

plt.imshow(img)

plt.title(f"Original: {original\_label}")

plt.axis('off')

plt.subplot(1, 2, 2)

adv\_display = np.clip((adv\_x[0] + 1) \* 127.5, 0, 255).astype("uint8")  # from [-1,1] to [0,255]

plt.imshow(adv\_display)

plt.title(f"Adversarial: {adv\_label}")

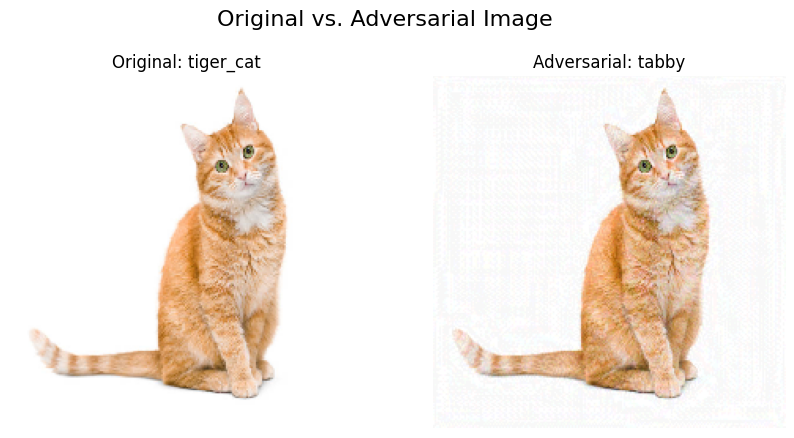
plt.axis('off')

plt.show()

print("Attack successful?", original\_label != adv\_label)

1. Click on the ***folder icon*** and upload the image of the cat from your device.
2. Now click on ***Run All*** or ***Ctrl + F9*** to run all the cells

**Output:**

****

**Explanation**

We used CleverHans to generate an adversarial example from an image of a cat. Let us understand the output:

**What the Images Show:**

* ***Original Image:*** The model predicted: tiger\_cat
* ***Adversarial Image:*** The model was fooled to predict: tabby

Both images look the same to a human eye (same orange cat, same pose), but a tiny, invisible noise was added to the adversarial image using the Fast Gradient Sign Method (FGSM).

***In this case:***

* Before attack: "tiger\_cat"
* After attack: "tabby"

Since they are different, the attack was successful.

**Try on your own:**

If you want to try, when will the model fail - Make epsilon extremely small so that attack is weak.

Change the epsilon or ***eps*** from ***0.05*** to ***1e - 6***