**ECE579\_Assignment2**

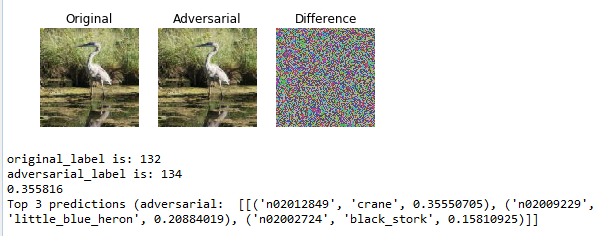
**Xin Xu**

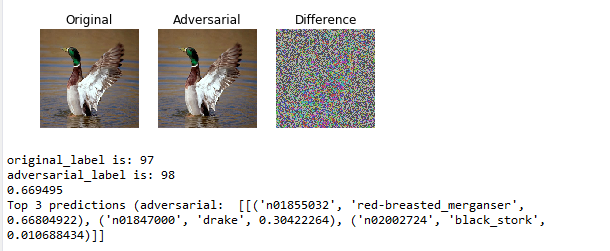
**Picture:** 10 pictures in ‘Aquatic bird’ of ‘Bird’ in Imagenet.

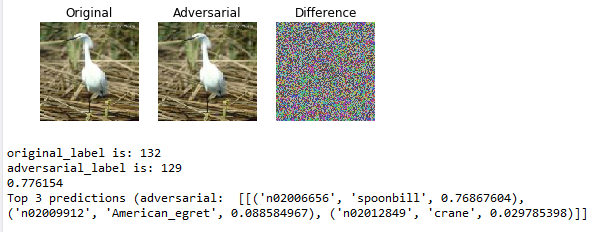
**Criteria:** Misclassification. And I decide to show the top 3 predictions and their possibilities.

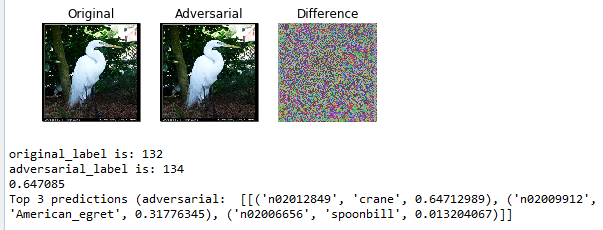
**Attack:**

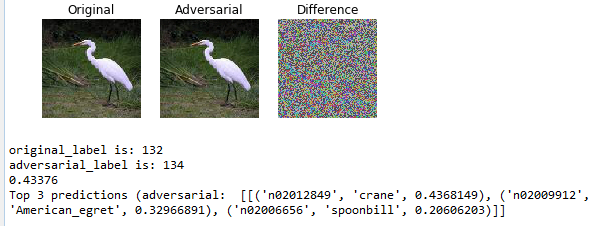
1. FGSM: (with the possibility of the prediction in third row)

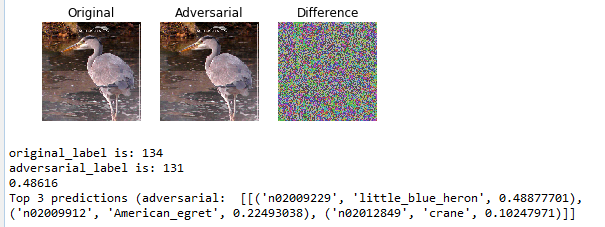
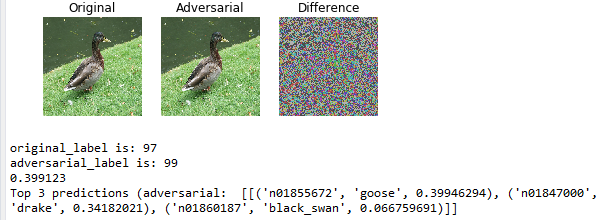


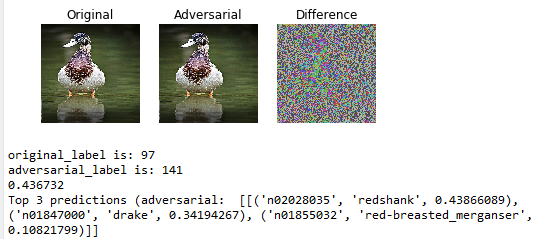


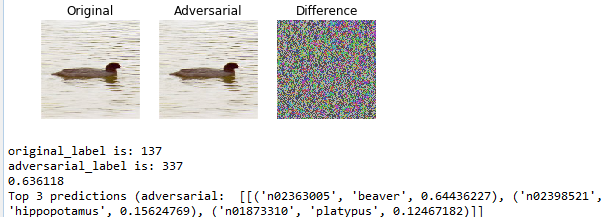


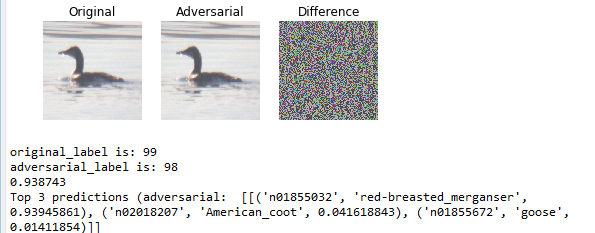




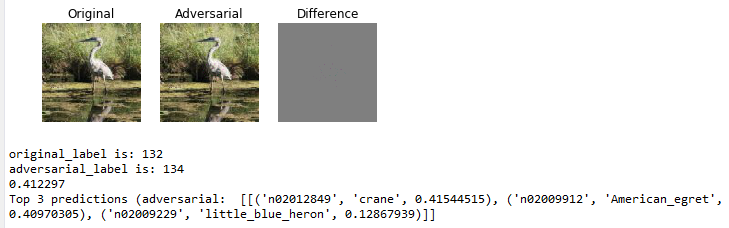
  


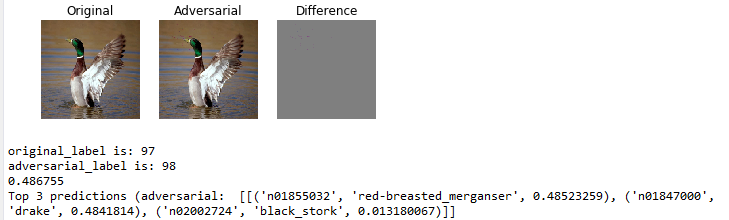


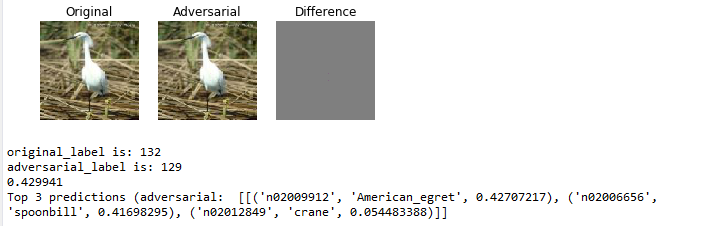


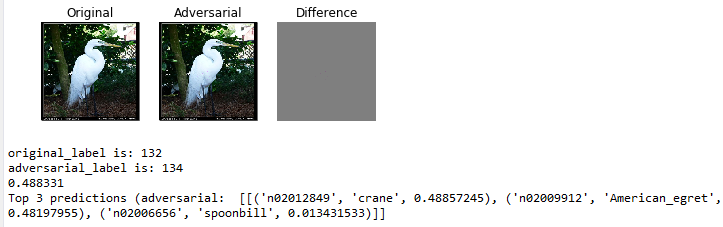


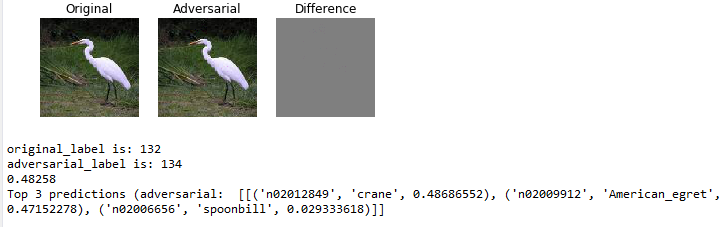
1. SaliencyMapAttack(with the possibility of the prediction in third row)

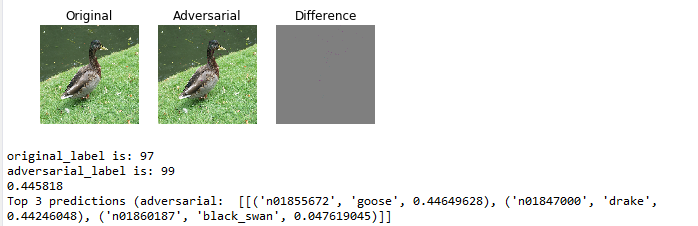


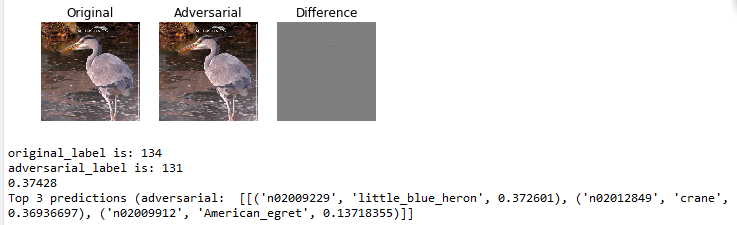


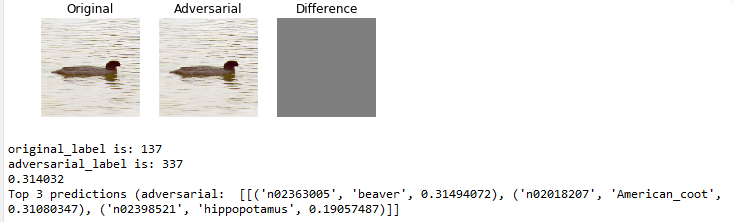


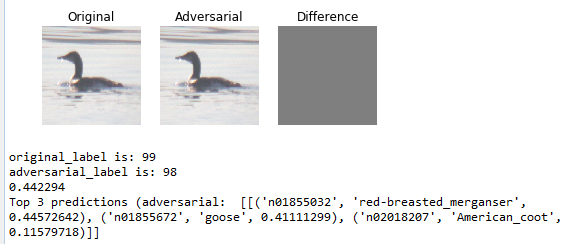






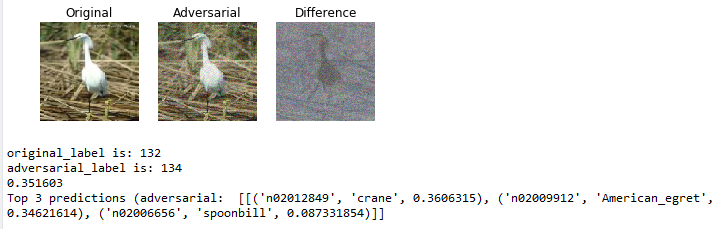




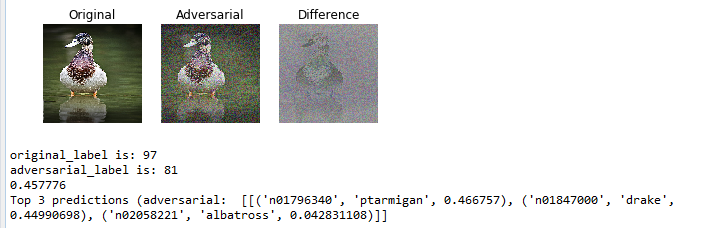
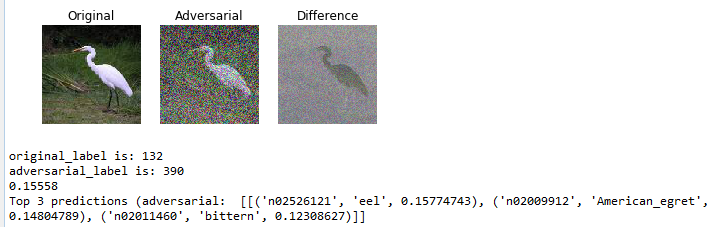


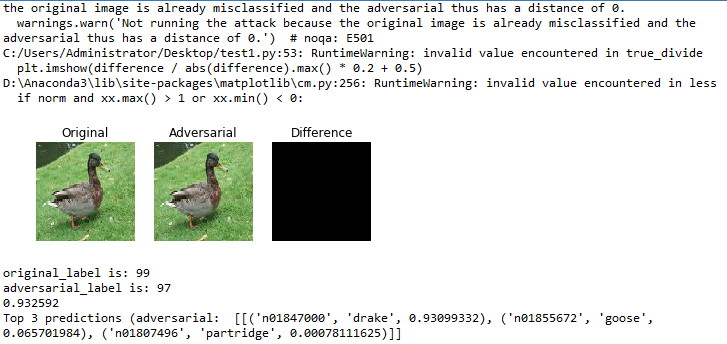
1. BlendedUniformNoiseAttack(with the possibility of the prediction in third row)

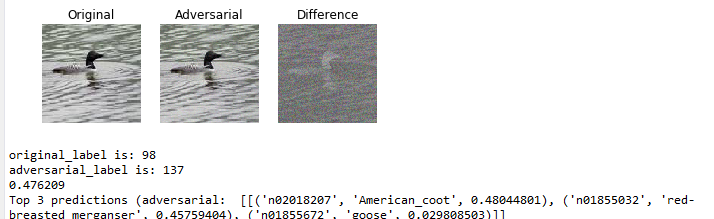








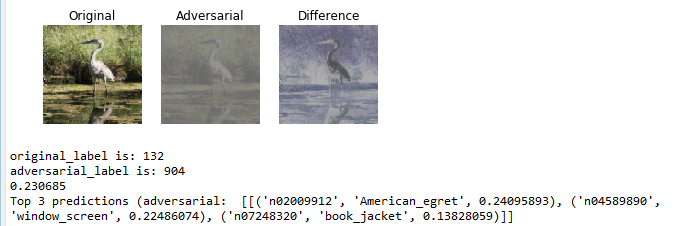


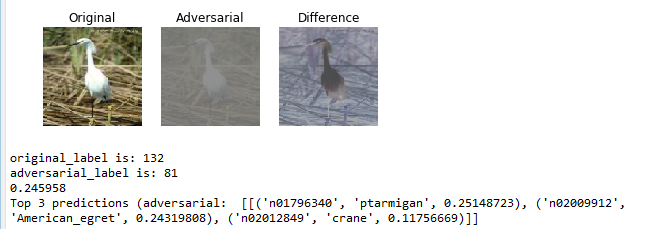




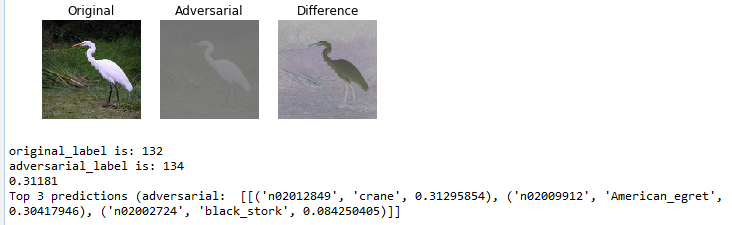


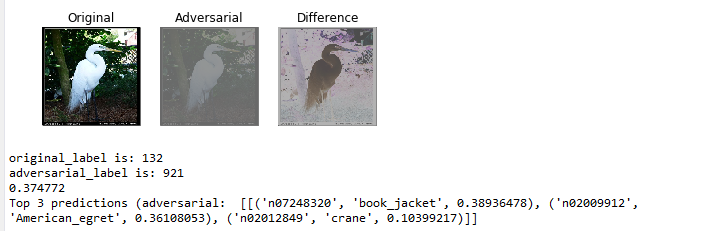
1. ContrastReductionAttack(with the possibility of the prediction in third row)

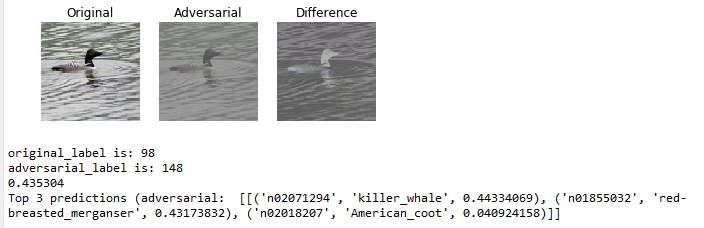


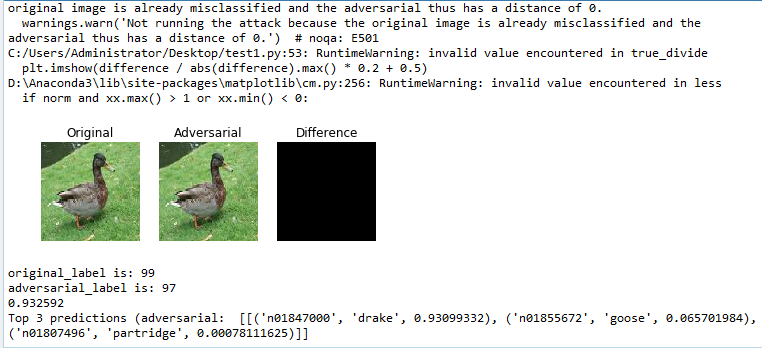


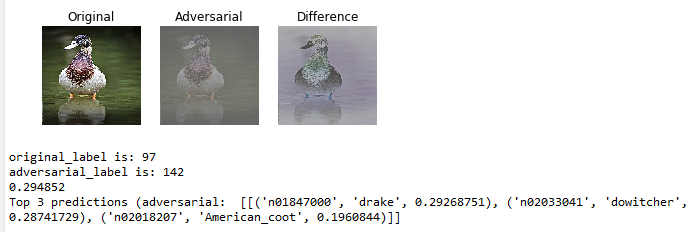


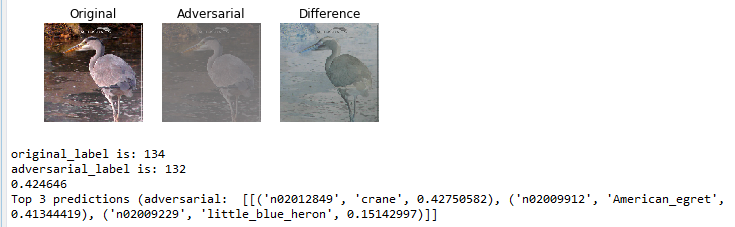


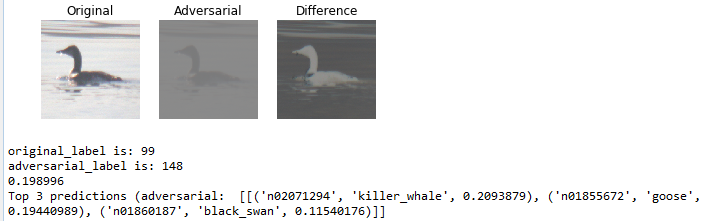












**Code:**

import foolbox

import keras

import numpy as np

from keras.applications.resnet50 import ResNet50

from PIL import Image

***from foolbox.criteria import Misclassification***

***from keras.applications.resnet50 import preprocess\_input***

***from keras.applications.resnet50 import decode\_predictions***

# instantiate model

***keras.backend.set\_learning\_phase(0)***

***kmodel = ResNet50(weights='imagenet')***

***preprocessing = (np.array([104, 116, 123]), 1)***

***fmodel = foolbox.models.KerasModel(kmodel, bounds=(0, 255), preprocessing=preprocessing)***

***# get source image and label***

***image = Image.open('C:/Users/Administrator/Desktop/n01844917/n01844917\_1145.jpeg')***

***shape=(224,224)***

***image=image.resize(shape)***

***image = np.asarray(image, dtype=np.float32)***

***image = image[:, :, :3]***

***assert image.shape == shape + (3,)***

***label = np.argmax(fmodel.predictions(image))***

***criterion = Misclassification()***

# apply attack on source image

# ::-1 reverses the color channels, because Keras ResNet50 expects BGR instead of RGB

***attack = foolbox.attacks.ContrastReductionAttack(fmodel,criterion)***

adversarial = attack(image[:, :, ::-1], label)

import matplotlib.pyplot as plt

plt.figure()

plt.subplot(1, 3, 1)

plt.title('Original')

plt.imshow(image / 255) # division by 255 to convert [0, 255] to [0, 1]

plt.axis('off')

plt.subplot(1, 3, 2)

plt.title('Adversarial')

plt.imshow(adversarial[:, :, ::-1] / 255) # ::-1 to convert BGR to RGB

plt.axis('off')

plt.subplot(1, 3, 3)

plt.title('Difference')

difference = adversarial[:, :, ::-1] - image

plt.imshow(difference / abs(difference).max() \* 0.2 + 0.5)

plt.axis('off')

plt.show()

***ad\_label=np.argmax(fmodel.predictions(adversarial))***

***print("original\_label is:",label)***

***print("adversarial\_label is:",ad\_label)***

***print(foolbox.utils.softmax(fmodel.predictions(adversarial))[ad\_label])***

***adversarial\_rgb = adversarial[np.newaxis, :, :, ::-1]***

***preds = kmodel.predict(preprocess\_input(adversarial\_rgb.copy()))***

***print("Top 3 predictions (adversarial: ", decode\_predictions(preds, top=3))***