Mira Plante
Adversarial Crafting
Full Code Repository with Resources [link]

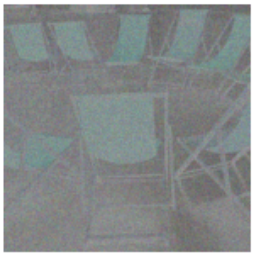
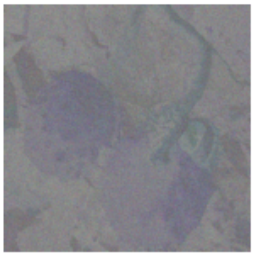
Over the course of the assignment, modifications to the foolbox tool had to be made to correctly print out the original image, the adversarial image, and the difference between the two. Code was also written present in the repository to convert the numerical labels for imagenet to human-readable text labels.

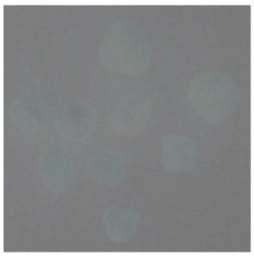
LinearSearchBlendedUniformNoiseAttack (Blended Uniform Noise Attack)
This attack was decently successful, lowering the accuracy to 0%, however this attack wasn't 100% successful because the convertible image label remained the same after the attack. The main parameters modified here were distance and the epsilon values.

Code for the Attack:

```
import foolbox as fb
import tensorflow as tf
import eagerpy as ep
import numpy as np

# model creation based on example given by tool creator
model = tf.keras.applications.ResNet50(weights="imagenet")
preprocessing = dict(flip_axis=-1, mean=[103.939, 116.779, 123.68])
bounds = (0, 255)
fmodel = fb.TensorFlowModel(model, bounds=bounds,
preprocessing=preprocessing)
fmodel = fmodel.transform_bounds((0, 1))

# image obtaining and processing
images, labels = fb.utils.samples(fmodel, dataset='imagenet',
batchsize=10)
images = ep.astensor(images)
labels = ep.astensor(labels)
criterion = fb.criteria.Misclassification(labels)

# actual attack
distance = fb.distances.LpDistance(50.0)
attack =
fb.attacks.LinearSearchBlendedUniformNoiseAttack(distance=distance)
raw, clipped, is_adv = attack(fmodel, images, criterion=criterion,
```

```
epsilons=0.75)

# show three images using modified tool function (found in code repo)
print("LinearSearchBlendedUniformNoiseAttack ACCURACY: ",
fb.utils.accuracy(fmodel, clipped, labels))
difference = clipped - images
difference = difference / abs(difference).max() * 0.2 + 0.5
imagesFunction(images, clipped, difference)

# printing out human-readable text labels
images_, labels_ = ep.astensors(clipped2, labels)
predictions = fmodel(images_).argmax(axis=-1)

printStatement = ""
for i in predictions.numpy():
  printStatement += d[i] + " | "
print(predictions.numpy(), " = ", printStatement)
```

Visualization of Adversarial Changes to Image	OG Label	New Label
Original  Adversarial  Difference 	'bullmastiff'	'Borderterrier'
Original  Adversarial  Difference 	'foldingchair'	'apron'
Original  Adversarial  Difference 	'beaker'	'soapdispenser'
Original  Adversarial  Difference 	'buckeye,horsechestnut,conker'	'jackfruit,jak,jack'

Original  Adversarial  Difference 	'strawberry'	'pomegranate'
Original  Adversarial  Difference 	'thatch,thatchedroof'	'mobilehome,manufacturedhome'
Original  Adversarial  Difference 	'jeep,landrover'	'racer,racecar,racingcar'
Original  Adversarial  Difference 	'convertible'	'convertible'

Original  Adversarial  Difference 	'yurt'	'mountaint ent'
Original  Adversarial  Difference 	'bottlecap'	'jellyfish'

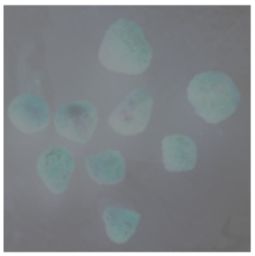
LinearSearchContrastReductionAttack (Contrast Reduction)
This attack was the most difficult attack to execute successfully and required the most parameter changes and optimizations to get any changes to accuracy on the model, with the two parameters being epsilons (this attack had to have a high epsilon value) and distance. Two other contrast reduction attacks, *L2ContrastReductionAttack* and *BinarySearchContrastReductionAttack*, were also tested with the linear search option making the most impact on images. The code and examples of images below show how the attack changed images, with the convertible image and label not changing between the original image and the adversarial image. This attack lowered the accuracy of the model to 0% from 90%, however as the label for the convertible didn't change at all, this attack is not 100% successful in changing all labels of the images.

Code for the Attack:

```
import foolbox as fb
import tensorflow as tf
import eagerpy as ep
import numpy as np

# model creation based on example given by tool creator
model = tf.keras.applications.ResNet50(weights="imagenet")
preprocessing = dict(flip_axis=-1, mean=[103.939, 116.779, 123.68])
bounds = (0, 255)
fmodel = fb.TensorFlowModel(model, bounds=bounds,
preprocessing=preprocessing)
fmodel = fmodel.transform_bounds((0, 1))

# image obtaining and processing
images, labels = fb.utils.samples(fmodel, dataset='imagenet',
batchsize=10)
images = ep.astensor(images)
labels = ep.astensor(labels)
criterion = fb.criteria.Misclassification(labels)

# actual attack
distance = fb.distances.LpDistance(50.0)
attack =
fb.attacks.LinearSearchContrastReductionAttack(distance=distance)
raw2, clipped2, is_adv2 = attack(fmodel, images, criterion=criterion,
epsilons=0.75)
```

```
# image obtaining and processing
print("L2ContrastReductionAttack ACCURACY: ", fb.utils.accuracy(fmodel,
clipped2, labels))
difference = clipped2 - images
difference = difference / abs(difference).max() * 0.2 + 0.5
imagesFunction(images, clipped2, difference)

# printing out human-readable text labels
images_, labels_ = ep.astensors(clipped2, labels)
predictions = fmodel(images_).argmax(axis=-1)

printStatement = ""
for i in predictions.numpy():
  printStatement += d[i] + " | "
print(predictions.numpy(), " = ", printStatement)
```

Visualization of Adversarial Changes to Image	OG Label	New Label
Original Adversarial Difference 	'bullmastiff'	Staffordshirebullterrier,Staffordshirebullterrier'
Original Adversarial Difference 	'foldingchair'	'radiotelescope,radioreflector'
Original Adversarial Difference 	'beaker'	'nipple'
Original Adversarial Difference 	'buckeye,horsechestnut,conker'	'custardapple'

Original  Adversarial  Difference 	'strawberry'	'fig'
Original  Adversarial  Difference 	'thatch,thatchedroof'	'mobilehome,manufacturedhome'
Original  Adversarial  Difference 	'jeep,landrover'	'towtruck,towcar,wrecker'
Original  Adversarial  Difference 	'convertible'	'convertible'

Original  Adversarial  Difference 	'yurt'	'picketfence,paling'
Original  Adversarial  Difference 	'bottlecap'	'puck,hockeypuck'

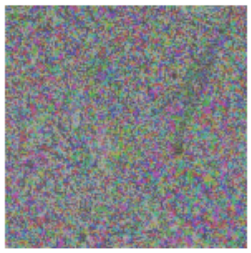
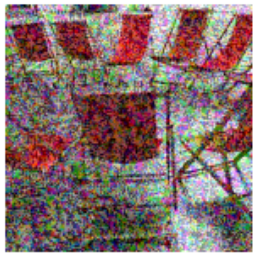
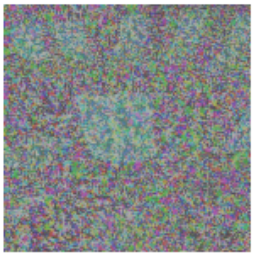
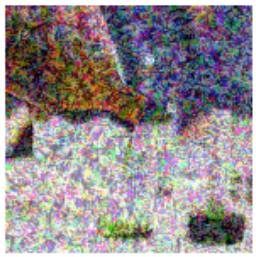
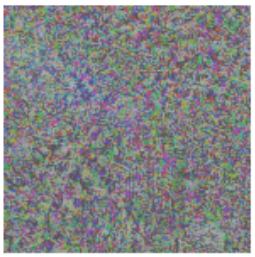
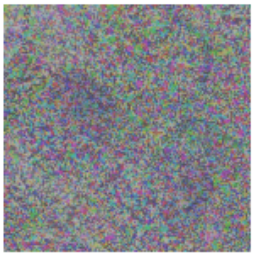
LinfFastGradientAttack (FGSM) Attack

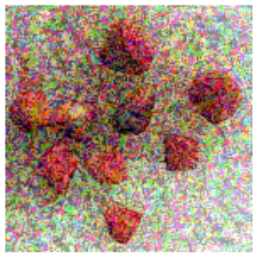
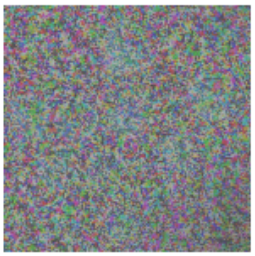
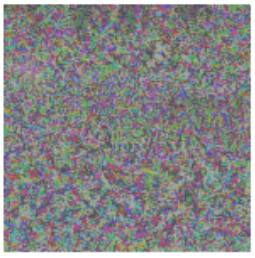
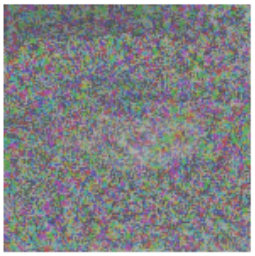
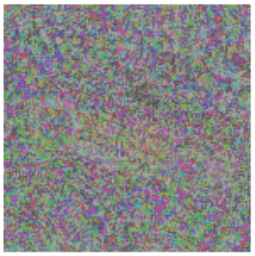
This attack seemed to be the most successful out of the three, as it lowered the accuracy from 90% to 0% and changed every label given in the sample image set. The code for the attack is below, as is the output for the regular and adversarial models alongside the changed labels. The parameter random_start was set to True and False, with True being the decided parameter with the best accuracy change.

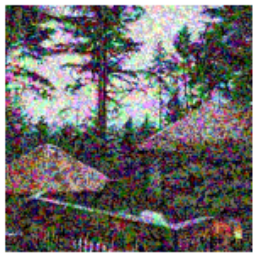
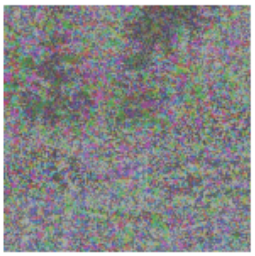
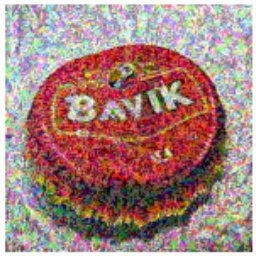
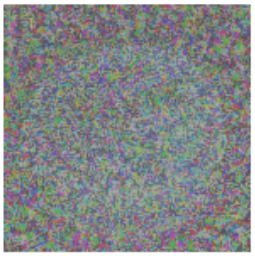
Code for the Attack:

```
import foolbox as fb
import tensorflow as tf
import eagerpy as ep
import numpy as np

# model creation based on example given by tool creator
model = tf.keras.applications.ResNet50(weights="imagenet")
preprocessing = dict(flip_axis=-1, mean=[103.939, 116.779, 123.68])
bounds = (0, 255)
fmodel = fb.TensorFlowModel(model, bounds=bounds,
preprocessing=preprocessing)
fmodel = fmodel.transform_bounds((0, 1))

# image obtaining and processing
images, labels = fb.utils.samples(fmodel, dataset='imagenet',
batchsize=10)
images = ep.astensor(images)
labels = ep.astensor(labels)
criterion = fb.criteria.Misclassification(labels)

# actual attack
attack = fb.attacks.LinfFastGradientAttack(random_start=True)
raw3, clipped3, is_adv3 = attack(fmodel, images, criterion=criterion,
epsilons=0.3)

# image obtaining and processing
print("LinfFastGradientAttack ACCURACY: ", fb.utils.accuracy(fmodel,
clipped3, labels))
difference = clipped3 - images
difference = difference / abs(difference).max() * 0.2 + 0.5
imagesFunction(images, clipped3, difference)
```

```
# printing out human-readable text labels
images_, labels_ = ep.astensors(clipped3, labels)
predictions = fmodel(images_).argmax(axis=-1)

printStatement = ""
for i in predictions.numpy():
  printStatement += d[i] + " | "
print(predictions.numpy(), " = ", printStatement)
```

Visualization of Adversarial Changes to Image	OG Label	New Label
Original  Adversarial  Difference 	'bullmastiff'	'braincoral'
Original  Adversarial  Difference 	'foldingchair'	'confectionery,confectionary,candystore'
Original  Adversarial  Difference 	'beaker'	'braincoral'
Original  Adversarial  Difference 	'buckeye,horsechestnut,conker'	'chain'

Original  Adversarial  Difference 	'strawberry'	'braincoral'
Original  Adversarial  Difference 	'thatch,thatchedroof'	'pinwheel'
Original  Adversarial  Difference 	'jeep,landrover'	'bubble'
Original  Adversarial  Difference 	'convertible'	'bubble'

Original  Adversarial  Difference 	'yurt'	'parkbench'
Original  Adversarial  Difference 	'bottlecap'	'purse'