Adversarial Machine Learning

Homework Assignment 2

The assignment is due by the end of the day on Tuesday, February 27.

Objective:

Implement black-box evasion attacks against deep learning-based classification models.

Part 1: Boundary Attack

The Boundary Attack is a black-box evasion attack based on the paper by Brendel et al. (2018), which we covered in Lecture 5. The boundary attack uses only the final predicted label by a black-box model to create adversarial samples, i.e., it is a decision-based attack. The following notebook in the Adversarial Robustness Toolbox explains the implementation of the boundary attack on ImageNet images. In addition, an implementation of the Boundary Attach by a student from last year's offering of the AML course can be found at this link.

Dataset: We will use the GTSRB (German Traffic Sign Recognition Benchmark) dataset. The dataset consists of about 51,000 images of traffic signs. There are 43 classes of traffic signs, and the size of the images is 32×32 pixels. The distribution of images per class is shown in Figure 1. More information about the dataset can be found at this link.

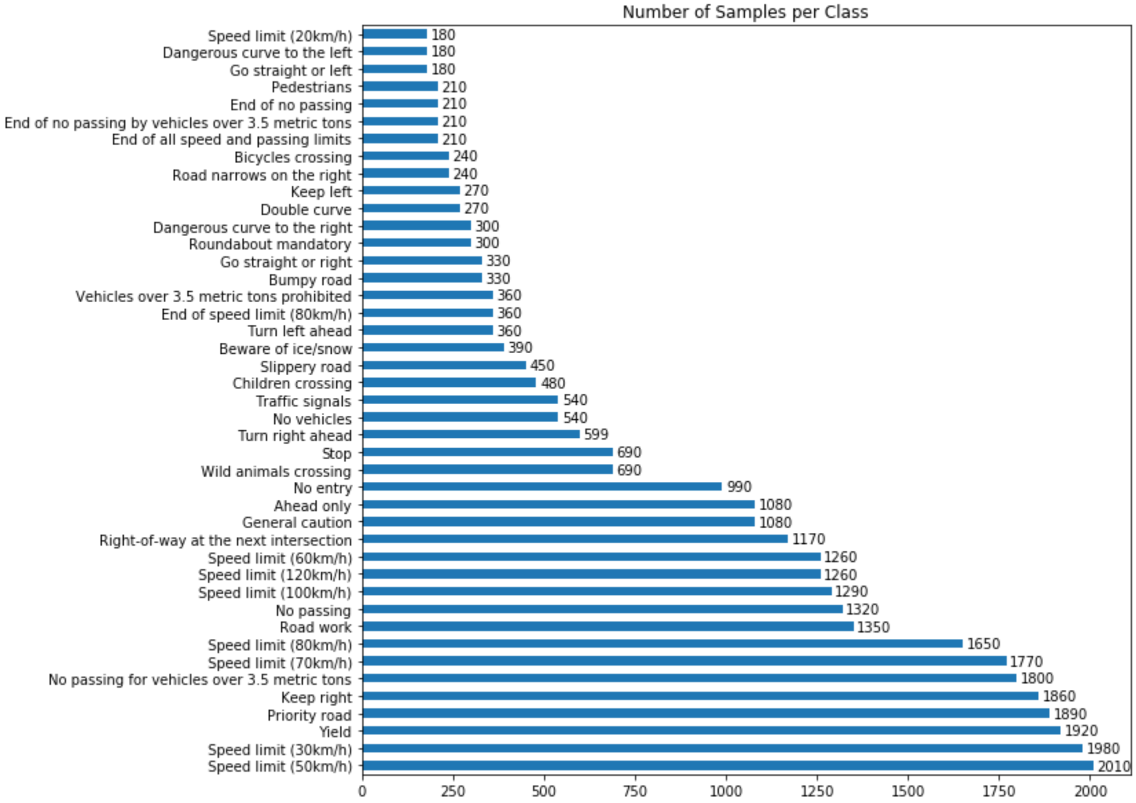


Figure 1. Images per class in GTSRB dataset.

Task 1: Train a convolutional NN for classification of the traffic signs in the dataset.

Use the provided Data Loader file to load the dataset.

The recommended CNNs for this task are either VGG-16 or ResNet-50, but implementing any other CNN is acceptable.

Perform hyperparameter tuning to obtain accuracy on the test dataset above 95%.

Estimated running time: between 5 and 15 minutes.

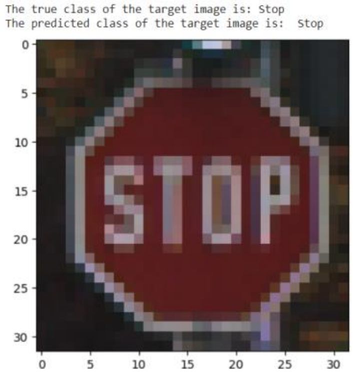
Report (20 marks): (a) Report the classification accuracy for the train set, validation set, and test set of images. For full marks, it is expected to report test accuracy above 95%. Plot the training and validation loss and accuracy curves. If applicable, provide any other observations regarding the training of the model.

Task 2: Implement an untargeted boundary attack against the trained model.

You can use the solution by the student from last year's AML course found at this link as a guidance for implementing the attack.

Important note: note that in the student's solution the images files are in the range from 0 to 255 pixels, and to display the images in Matplotlib, the student used the following lines in all instances `plt.imshow(image.astype(np.uint))`. In our case, the images files are scaled in the range from 0 to 1, and there is no need to use `astype(np.uint)` for displaying the images. If you use the student's code, make sure to remove `astype(np.uint)`.

Step 1: Select the image with index 111 from the test dataset to be used for creating an adversarial sample. It is a Stop Sign image. First, make sure that the DL classifier correctly predicts the class of the image. To check it, plot the image with the ground truth label and the predicted label by the DL model.



```
The true class of the target image is: Stop
The predicted class of the target image is:  Stop
```

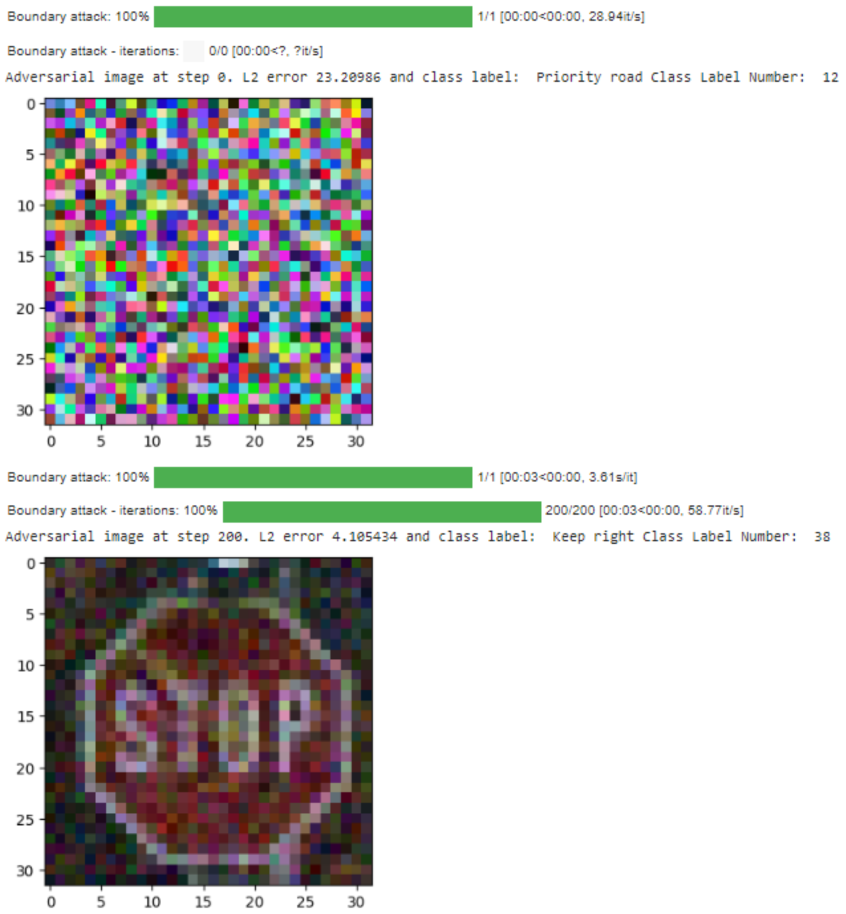
Figure 2. Selected image for the attack.

Step 2: Using the boundary attack, create an adversarial image that will change the label of the selected original image. You can use similar parameters for the attack as in the student's solution or as in it listed example notebook in the ART toolbox, or if you wish you can adopt different parameters. Print the L2 norm and the label of the adversarial image for each step of the attack, similar to Figure 3.

Step 3: Plot the final adversarial image with the predicted label by the classifier.

Estimated time: between 2 and 10 minutes.

Report (20 marks): (a) Plot the required figures in Steps 1 to 3. (b) Write between 5 and 10 sentences to explain the boundary attack.



```
Boundary attack: 100%  1/1 [00:00<00:00, 28.94it/s]
Boundary attack - iterations:  0/0 [00:00<?, ?it/s]
Adversarial image at step 0. L2 error 23.20986 and class label:  Priority road Class Label Number:  12
```

```
Boundary attack: 100%  1/1 [00:03<00:00, 3.61s/it]
Boundary attack - iterations: 100%  200/200 [00:03<00:00, 58.77it/s]
Adversarial image at step 200. L2 error 4.105434 and class label:  Keep right Class Label Number:  38
```

Figure 3. Untargeted boundary attack.

Task 3: Implement a targeted boundary attack against the trained model.

You can again use the codes in the student's solution as guidance.

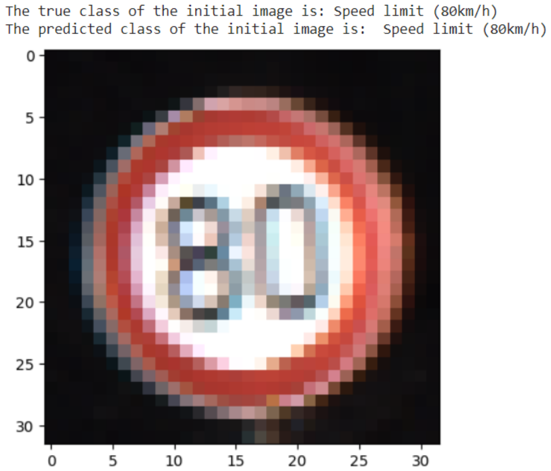
The goal will be to misclassify the same Stop Sign from Task 2 as the target class Speed Limit (80km/h).

Step 1: Select the Stop Sign image with index 111 from the test dataset to be used for creating an adversarial sample.

Step 2: Select the Speed Limit (80km/h) image with index 217 from the test dataset with the target class label. Plot the image with the ground truth label and the predicted label by the DL model, as in Figure 4.

Step 3: Using the boundary attack, create an adversarial image that will change the label of the selected image to the target label Speed Limit (80km/h). Print the L2 norm and the label of the adversarial sample for each step of the attack, similar to Figure 3.

Step 4: Plot the adversarial image with the predicted label by the classifier.



```
The true class of the initial image is: Speed limit (80km/h)
The predicted class of the initial image is:  Speed limit (80km/h)
```

Figure 4. Selected image from the target class label.

Estimated time: between 2 and 10 minutes.

Report (20 marks): (a) Plot the required figures in Steps 1 to 4. (b) Briefly describe the targeted boundary attack.

Part 2: Transferability Attack

Create a substitute model for celebrity recognition, and transfer adversarial samples to a corresponding model hosted by Clarifai. This is a black-box attack, because we don't have access to the model hosted by Clarifai.

Task 1: Train a deep-learning Vision Transformer model for classification of images of celebrities.

Dataset: We will use a dataset of celebrity faces, which is a subset of a larger dataset called LFW (Labeled Faces in the Wild). The images are collected from the web, and are labeled with the name of the person. The dataset for this assignment consists of 5,113 images of 62 celebrities. The images have only the face of the person cropped out from the original images. Sample images are shown in Figure 5.

Use the provided Data Loader file to load the dataset.

For the Vision Transformer, use the provided code named ViT_PyTorch as a guidance for training the model. The code employs a Vision Transformer (ViT) with a Linformer backbone. You should be able to use the same model with just a minor modification for the number of classes.

Perform hyperparameter tuning (learning rate and number of epochs) to obtain accuracy on the test dataset above 80%.

Estimated running time: between 5 and 20 minutes.

Report (10 marks): (a) Report the classification accuracy for the train set, validation set, and test set of images. For full marks, it is expected to report test accuracy above 80%. Plot the training and validation loss and accuracy curves. If applicable, provide any other observations regarding the training of the model.

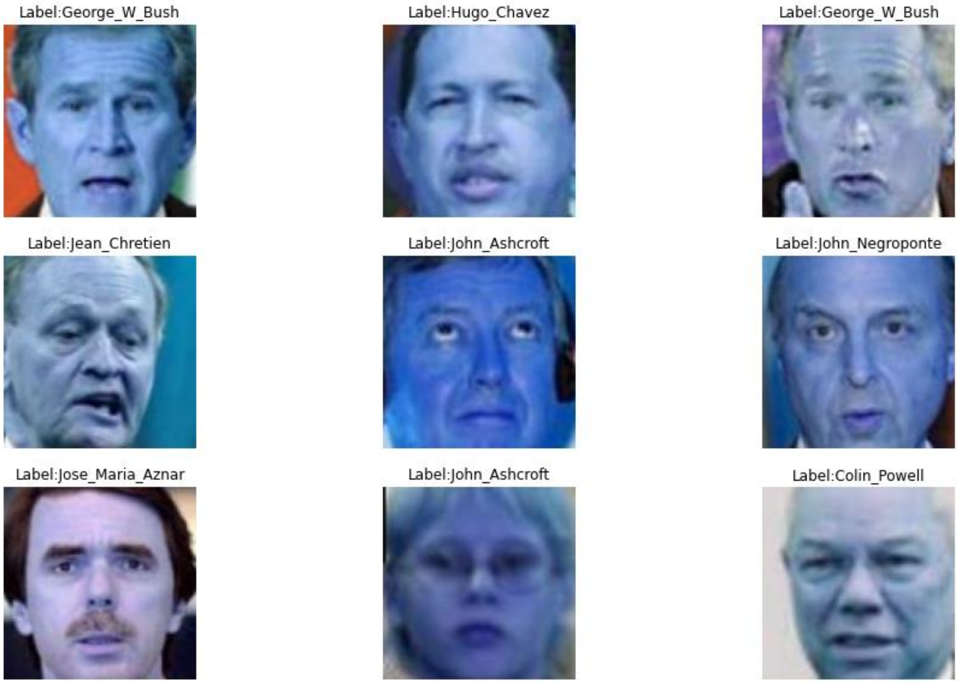
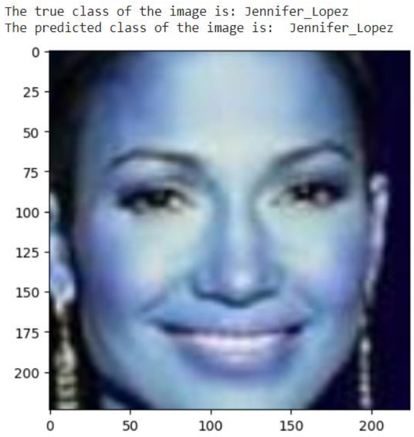


Figure 5. Sample images from the celebrity faces dataset.

Task 2: Create adversarial samples against Clarifai's web ML model for celebrity recognition.

Step 1: Select the image of Jennifer Lopez with index 343 from the test dataset for creating adversarial samples. Plot the image with the ground truth label and the predicted label by the DL model.



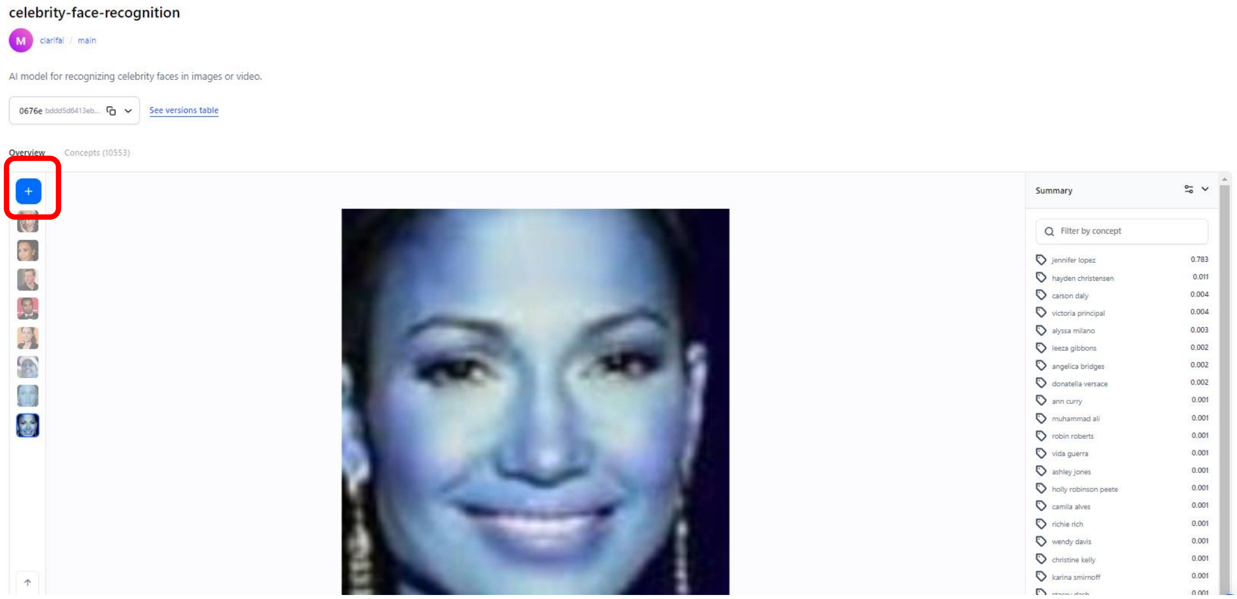
```
The true class of the image is: Jennifer_Lopez
The predicted class of the image is:  Jennifer_Lopez
```

Figure 6. Selected image for the attack.

Step 2: Visit the Clarifai website: https://www.clarifai.com/models/celebrity-face-recognition

You will be prompted to sign in, and I believe that the easiest way is to sign in with an existing Gmail account or GitHub account.

The API is shown in Figure 7. Click on the "**+**" button and select "**Try your own image or video**" to upload the selected image in Step 1.



AI model for recognizing celebrity faces in images or video.

jennifer lopez	0.783
hayden christensen	0.011
carson daly	0.004
victoria principal	0.004
alyssa milano	0.003
leeza gibbons	0.002
angelica bridges	0.002
donatella versace	0.002
ann curry	0.001
muhammad ali	0.001
robin roberts	0.001
vida guerra	0.001
ashley jones	0.001
holly robinson peete	0.001
camila alves	0.001
richie rich	0.001
wendy davis	0.001
christine kelly	0.001
karina smirnoff	0.001
stacey dash	0.001

Figure 7. Clarifai API for celebrity recognition.

Note in Figure 7 that the model correctly classified the image of Jennifer Lopez with a high confidence of 0.783.

Step 3: Apply the PGD attack to create non-targeted adversarial sample for the image from Step 1.

In PyTorch, you can create a batch from one image using:

```
sub_dataset = Subset(test_dataset, [343])
subsest_dataloader = DataLoader(sub_dataset, batch_size=1)
```

You can save the adversarial sample with the following code:

```
import torchvision.transforms.functional as TF
from torchvision.utils import save_image
TF.to_pil_image(adversarial_image[0].cpu()).save('im1.jpg')
```

Upload the adversarial sample to the Clarifai's website to check if it is misclassified.

Find the minimum perturbation level for the image to be misclassified by the Clarifai's model.

Plot the adversarial image with the lowest perturbation.

Step 4: Repeat the same steps for the image of David Beckham with index 442 from the test dataset. Find the minimum perturbation level for the image to be misclassified by the Clarifai's model.

Step 5: Repeat the same steps for the image of Winona Ryder with index 53 from the test dataset. Find the minimum perturbation level for the image to be misclassified by the Clarifai's model.

Estimated running time: between 5 and 10 minutes.

Report (30 marks): Plot the original images and the ground-truth label, the adversarial images, the predictions by the Clarifai's model, and state the applied level of perturbation. Provide a brief discussion of your opinion of the target model, and whether you found it easy or difficult to perform the attacks.

Bonus (10 marks): Use the PGD attack to create targeted adversarial samples. You can use either the same images, or you can select other images from the test dataset. Feel free to select a target label as you wish.

Submission documents

1. All notebooks and a brief report with tables, graphs, and results (either in MS Word/PDF or inserted inline in the Jupyter notebooks).