Part 1

The task for Part 1 was to create transformation defenses for a naïve PGD attack. As a baseline, the defenseless model was evaluated on benign and adversarial images generated with the provided PGD function. The model was also evaluated on clean and adversarial images with three input image transformation defenses: JPEG compression, resizing, and Gaussian blur. The recorded metrics for the baseline and each defense were accuracy on clean images (1), accuracy on adversarial images (2), and success rate of adversarial images (3):

$$\frac{count(benign\ images\ classified\ as\ y_{true})}{count(benign\ images\ processed)} \tag{1}$$

$$\frac{count(AEs\ classified\ as\ y_{true})}{count(AEs\ processed)} \tag{2}$$

$$\frac{count\left(AEs\ classified\ as\ y_{target}\right)}{count(AEs\ processed)} \tag{3}$$

The test was run repeatedly over the same dataset with the same model and attack, but different transformation parameters. The ideal parameters were compression quality of 28, 13 pixels for resizing, and a 5x5 kernel with a standard deviation of 2 for Gaussian blur.

The results of the baseline and defense tests with the optimized transformation parameters are displayed in table 1.

Table 1. Accuracy scores per by defense with naïve PGD

Defense	Benign Accuracy (1)	AE Accuracy (2)	Attack Success (3)
None	77.00%	4.00%	93.00%
JPEG Compression	71.00%	35.00%	49.00%
Resizing	71.00%	61.00%	16.00%
Blurring	75.00%	55.00%	20.00%

The results show increases in AE accuracy and attack success percentages with slight decreases in benign accuracy between the baseline and the three defenses. More specifically, image resizing was the best defense, raising the AE accuracy by 57.00%, decreasing attack success by 77.00%, and decreasing benign accuracy by 6.00%. Gaussian blur performed the second best, raising AE accuracy by 51.00% and decreasing attack success by 73.00% with a benign accuracy decrease of 2.00%. Of the three defenses, JPEG compression performed the worst, increasing AE accuracy by 31.00%, decreasing attack success by 44.00%, and decreasing benign accuracy by 6.00%.

Part 2

The goal in Part 2 was to create an attack that succeeded better than PGD on the defenses implemented in Part 1. To break the transformation defense created in Part 1, an expectation over transformation attack was created, averaging the gradients across each of the three transformations. The attack was evaluated on the four defense cases used for evaluating Part 1 with the three accuracy scores. Since the hypothetical scenario indicates that the creator of the defense is also the creator of the attack, it was assumed that the attacker had white-box access to the defense parameters in addition to the model.

To create adversarial examples with EOT, differentiable versions of the three transformations (via Kornia*) were applied to each image. For each image, the loss was computed according to eq. 4:

$$J = H\left(y_{target}, f\left(t(x)\right)\right) \tag{4}$$

where $H(l, n)$ is the cross-entropy loss between label l and some output n, y_{target} is the target label for the AE, $f(x)$ is the model's evaluation of an input image x, and $t(x)$ is the transformation result of an input image x. Informed by the assumption that the attacker had white-box access to the defense parameters, the optimized parameters from Part 1 were also used in the transformations in the EOT attack.

Table 2. Accuracy scores per defense with EOT

Defense	Benign Accuracy (1)	AE Accuracy (2)	Attack Success (3)
None	77.00%	22.00%	73.00%
JPEG Compression	71.00%	23.00%	63.00%
Resizing	71.00%	23.00%	70.00%
Blurring	75.00%	12.00%	84.00%

The EOT attack performed best on the model with Gaussian blur with an 84.00% success rate. The attack had a 70.00% success rate on the model when it used image resizing, and a 63% success rate on the model using JPEG compression. Despite a 7% difference in attack success between JPEG compression and image resizing, the model predicts AEs with a 23.00% accuracy, suggesting that an untargeted attack might be more efficient on JPEG compression protected models. Additionally, the large difference between the attack success on Gaussian blur and the other defenses suggests that the attack may be overfitting slightly.

*Note that the JPEG compression algorithm provided by Kornia does not handle non-CPU devices. To circumvent this, a local patch was made in the submission document following this pull request.

Part 3

The task for Part 3 was to train a new, more robust model through defensive distillation with the original, undefended model as the teacher. A base distillation algorithm using KL loss was first implemented, following the general structure of this algorithm from the Pytorch documentation. The crucial aspect of the algorithm from the Pytorch documentation was that both the teacher and student logits were divided by the temperature before the softmax was applied. However, despite extensive testing with different temperatures and other hyperparameters (epoch count, learning rate, loss function, etc.), the attack accuracy on this student model never dropped below 95%. During this process, it was noted that removing division by temperature from the softmax of the student logits yielded lower attack accuracies without significantly impacting the classification accuracy. As such, a modified version of the original algorithm was adopted without any divisor for the student logits. This meant that the softmax of the student outputs was being guided towards nearly equal probabilities with a slight peak in the logit corresponding to the teacher model's classification. To illustrate, the teacher output for one of the sample images with the correct label of 4 was

$$[1.8e\!-4 \quad 1.4e\!-4 \quad 3.2e\!-4 \quad 0.94 \quad 3.9e\!-4 \quad 5.8e\!-2 \quad 7.0e\!-4 \quad 5.8e\!-4 \quad 1.1e\!-4 \quad 1.2e\!-4]$$

while the corresponding student output was

$$[0.0993 \quad 0.0992 \quad 0.0998 \quad 0.1022 \quad 0.1001 \quad 0.1014 \quad 0.1002 \quad 0.0997 \quad 0.0992 \quad 0.0990]$$

After this modified algorithm was selected, iterative computations were performed (using a personal GPU) with a pre-selected dataset of length 100 on temperatures ranging from 1 to 150 with spaces of 5 in between. Through this process, it was determined that 90 was the optimal temperature for the dataset that was used.

Defensive distillation was evaluated using the same metrics from Parts 1 and 2, but the teacher accuracy on benign images was also recorded for comparison with that of the student model.

Table 3. Accuracy scores for distillation defense

Defense	Student Benign Accuracy (1)	Teacher Benign Accuracy (1)	AE Accuracy (2)	Attack Success (3)
Distillation	71.00%	77.00%	18.00%	63.00%

The distillation defense had a benign accuracy of 71.00%, 6.00% lower than that of the teacher. It also had 18.00% accuracy on AE classification, and lowered the attack success from 93.00% on the undefended teacher model in Part 1 to 63.00%. Though this was less efficient than transformation defenses from Part 1, it still tripled AE accuracy and lowered attack success by 30.00%.

This defense works by decreasing the difference between the model's output probabilities. Since PGD relies on output probabilities changing with perturbation, having high similarity between probability values significantly reduces the magnitude of the change in target class probability between iterations. This minimizes the gradient of the loss with respect to each pixel, making it much harder to find the correct direction to move the perturbation.