Adversarial Robustness of YOLOv8x-cls on CIFAR-100

Graduate Principles of AI — Term Project Report

Jeff M. Colfer
May 2025

1 Introduction

Image-classifying neural networks have become a silent work-horse of modern AI agents—routing autonomous vehicles, filtering social-media uploads, triaging radiographs, and guiding household robots. Their accuracy on public benchmarks has risen spectacularly; however, this success hides a brittle underbelly. Small, carefully sculpted perturbations that respect human perception can push predictions off a cliff. Quantifying that fragility, and the extent to which training choices mitigate it, is the central aim of this project.

I adopt YOLOv8x-cls, the extra-large variant of Ultralytics' 2023 YOLO-v8 family, chosen because it embodies many best practices of contemporary vision models (C2f residual blocks, SiLU activations, anchor-free head) while shipping a ready-made classification checkpoint. Using the CIFAR-100 dataset as a compact but diverse surrogate for natural images, I explore three training regimes:

(i) Clean-trained (400 untouched images)
(ii) Adversarial-trained (200 Projected Gradient Descent–generated images), and
(iii) Mixed (200 clean + 200 adversarial).

Each model is probed by four canonical evasion attacks—Fast Gradient Method (FGM), Projected Gradient Descent (PGD), Spatially Transformed perturbations, and the Square black-box attack—both with and without inexpensive inference-time defenses (JPEG compression plus 3-bit quantization). The experiment asks: How do training-data composition and lightweight defenses interact to shape robustness? The answers will inform future work that scales the analysis to ImageNet and the forthcoming YOLOv11 line.

2 Background

2.1 Image Classifiers: Architecture

Modern convolutional neural networks (CNNs) map an image tensor

$$x \in \mathbb{R}^{H \times W \times C}$$

successive layers:

$$y_{i,j,k} = \sigma(b_k + \sum_{m,n,c} W_{m,n,c} x_{i+m,\, j+n,\, c})$$

CNNs share parameters across spatial locations and extracting local patterns. Global Average Pooling (GAP) collapses each channel produces a translation-robust feature vector and reduces computational overhead. The fully-connected (FC) layer maps the features to class logits $z = Wg + b$. Soft-max then yields

$$p_k = \frac{e^{z_k}}{\sum_j e^{z_j}}$$

For a labelled sample (x, y) with a binary target label, the function to be minimized is the cross-entropy, given by:

$$-\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{C} y_{i,j}\log\left(p_{i,j}\right)$$

2.2 YOLOv8x-cls Backbone

YOLOv8x-cls repurposes the detection-oriented YOLOv8 backbone as a classifier. Key components:

(i) CSP-Darknet with C2f residual blocks for parameter-efficient feature extraction.
(ii) SPPF (Spatial-Pyramid-Pooling-Fast) that serially applies 5×5 max-pools to inject multi-scale context.
(iii) A minimalist classifier head GAP → FC → soft-max instead of bounding-box heads.

2.3 Evasion Attacks

Attack	Threat Model	Update Rule & Intuition
FGM / FGSM	White-box, 1-step $\ell\infty$	$x' = x + \varepsilon\cdot\text{sign}(\nabla_x L)$. A single linearized step exploits local gradient alignment.
PGD	White-box, multi-step $\ell\infty$	Iterates FGM with step α, projecting back into the ε-ball; strongest first-order adversary.
Spatial	White-box, geometric	Searches small rotations (±15°) and translations (±10 %). Exploits lack of perfect equivariance.
Square	Black-box, score-based	Adds localized square patches via random search using only logits queries; evades gradient masking.

Table 1—Evasion attack descriptions

Figure 1—A visual illustration of the input images produced by the attack methods. From top to bottom: FGM, PGD, Spatial, and Square methods.

2.4 Image Classifiers in Embodied AI Agents

In real-world systems (autonomous driving, biometric auth, medical imaging) such attacks can induce critical decisions on manipulated inputs, motivating robustness research. Autonomous vehicles, service robots, and AR devices rely on real-time image understanding. A single misclassification can erase pedestrians, mis-route drones, or unlock spoofed faces. Demonstrating digital vulnerabilities is therefore a first step toward hardening perception pipelines.

3 Methods

3.1 Experimental Environment

- Compute: Google Colab Pro+ session with an NVIDIA A100 40 GB GPU, CUDA 12, Python 3.10.
- Data: CIFAR-100 (50 000 train / 10 000 test, 32 × 32 RGB).

- Frameworks: Ultralytics 8.1, PyTorch 2.2, CleverHans 4.0, TorchAttacks 3.5, AutoAttack 2.1
- Artifacts: All trained models and logs pushed to a private Kaggle dataset for reproducibility.

3.2 Model Training

The pretrained yolov8x-cls.pt was loaded using the YOLO library from Ultralytics, and then trained on the CIFAR-100 dataset using the following training regimes

• Clean-only: 400 epochs, batch 128, cosine LR schedule.

• Adversarial-only: 200 epochs on PGD-8/255 samples (20 steps, $\alpha = 2/255$).

• Mixed: 200 epochs clean then 200 adversarial.

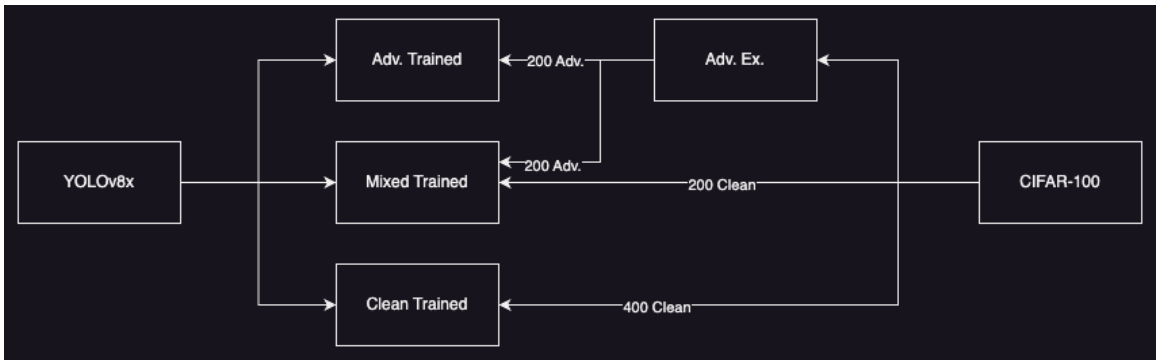

Figure 2—the training regime used to produce the three model versions.

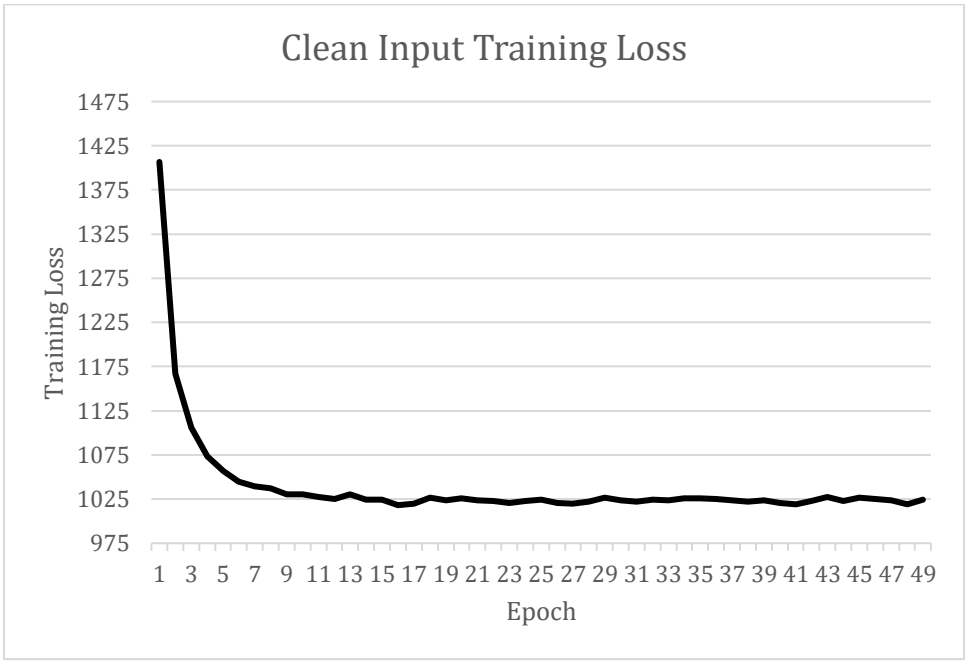

Figure 3—the training loss of the YOLOv8x-cls model trained for 400 epochs of clean inputs.

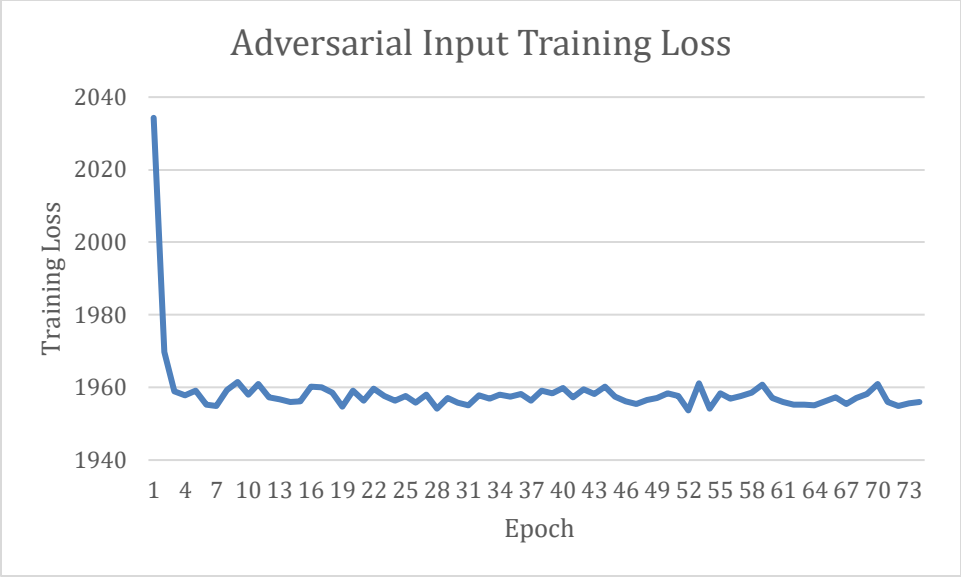

Figure 4— the training loss of the YOLOv8x-cls model trained for 200 epochs with generated adversarial inputs only

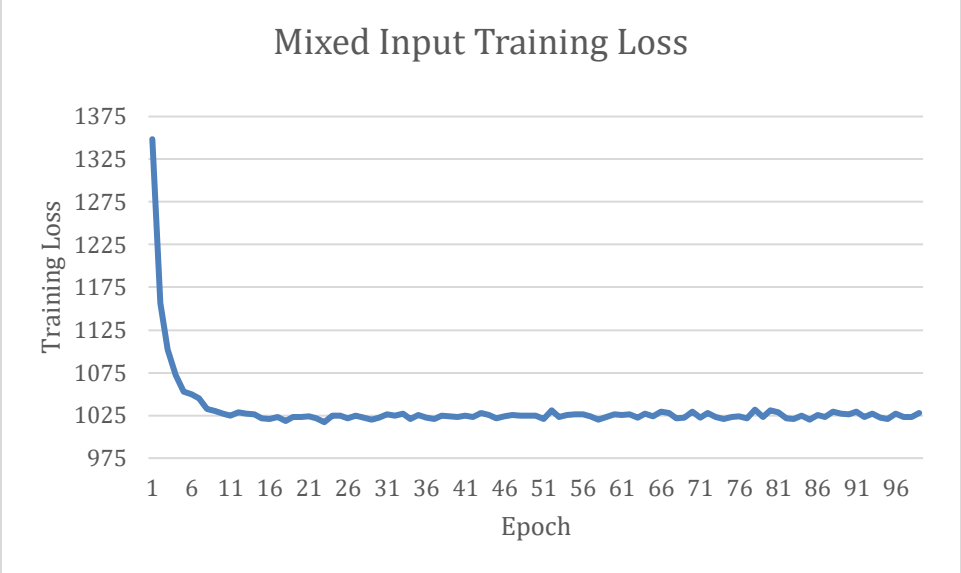

Figure 5—training loss of the YOLOv8x-cls model trained for 200 epochs with generated adversarial inputs

3.3 Defenses Against Evasion Attacks at Evaluation

3.3.1 Adversarial Training

The most important and effective defense against evasion attacks in general is adversarial training wherein the classifier's network is trained with inputs perturbed by each of the attack methods. The network learns *robust features*—ones that cannot be destroyed by small pixel noise. The trade-offs are that your training is much, much costlier: taking

roughly 20x longer in the case of this project. Also, model performance can be adversely affected resulting in decreased accuracy against clean image inputs.

In addition to adversarial training—by definition, performed during training of a network—inference defenses, such as JPEG compression and bit-depth reduction are also capable of providing defenses against gradient-based evasion attacks. These defenses are described below in Table 2.

JPEG Compression	**Bit-Depth Reduction**
• ***Pre-processing filter***	• ***"Feature Squeezing"***
• **RGB → YCbCr 4:2:0 → Block-DCT → aggressive quantization**	• **Quantize each channel to $n \leq 5$ bits**
• **Rounds away high-frequency coefficients where ε-scale noise hides**	• **Perturbations < ½ LSB vanish**
• **Decompress → forward pass**	• **Cuts gradient-based attacks (FGSM, PGD, Square)**
	• **Little help vs. spatial shifts/rotations**

Table 2—Inference defenses against evasion attacks.

3.4 Evaluation Strategy

The process of evaluating each of the three models resulting from the training strategy outlined in section 3.2 examines the performance of each against the four attacks outlined in section 2.3 by evaluating the precision, recall, F-1 score, accuracy and AOC-ROC of the models both with and without defenses. A logical diagram of this evaluation method is outlined in Figure 5 below.

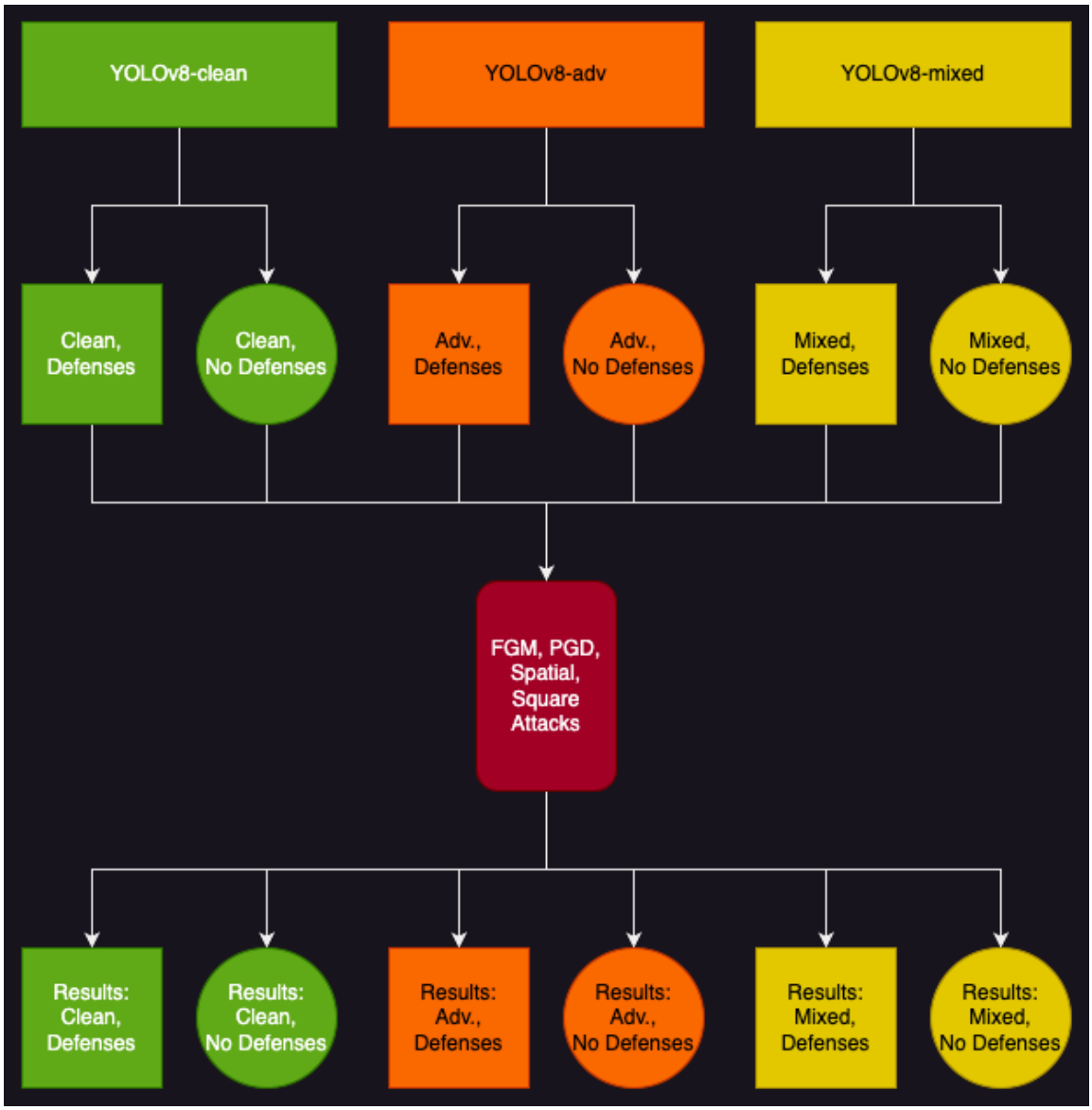

Figure 6—The performance of each model, with and without inference defenses, is evaluated using accuracy, precision, recall, F-1 score, and AOC-ROC.

3.5 Evaluation Metrics & Statistics

Top-1 accuracy with Wilson 95 % CIs was computed under all model-attack-defense scenario. For each model-attack pair a paired McNemar test compared undefended vs defended predictions with the null hypothesis that defenses do not improve the accuracy of the models

4 Results

4.1

Table 3 lists the performance of each model-attack combination with and without inference defenses.

Model	Defenses	Attack	Accuracy	Precision	Recall	F-1 Score	AOC-ROC
Clean-Trained	No	None	23.63	2.29E-01	2.36E-01	2.29E-01	6.14E-01
Clean-Trained	No	FGM	1.04	1.16E-02	1.04E-02	6.10E-03	5.00E-01
Clean-Trained	No	PGD	1.39	1.86E-02	1.39E-02	1.80E-02	5.02E-01
Clean-Trained	No	Spatial	17.32	1.76E-01	1.73E-01	1.67E-01	5.82E-01
Clean-Trained	No	Square	0	0.00E+00	0.00E+00	0.00E+00	4.95E-01
Clean-Trained	Yes	None	16.32	2.40E-02	1.82E-02	1.03E-02	5.77E-01
Clean-Trained	Yes	FGM	1.14	3.30E-02	1.14E-02	6.40E-03	5.01E-01
Clean-Trained	Yes	PGD	1.11	1.45E-02	1.11E-02	8.00E-03	5.01E-01
Clean-Trained	Yes	Spatial	11.6	1.32E-01	1.16E-01	1.11E-01	5.54E-01
Clean-Trained	Yes	Square	0	0.00E+00	0.00E+00	0.00E+00	4.95E-01
Adv-Trained	No	None	1.82	2.44E-02	1.82E-02	1.03E-02	5.04E-01
Adv-Trained	No	FGM	3.21	2.61E-02	3.21E-02	1.88E-02	5.11E-01
Adv-Trained	No	PGD	4.87	5.75E-02	4.87E-02	4.12E-02	5.20E-01
Adv-Trained	No	Spatial	1.93	2.63E-02	1.93E-02	1.17E-02	5.05E-01
Adv-Trained	No	Square	0	0.00E+00	0.00E+00	0.00E+00	4.95E-01

Adv-Trained	Yes	None	2.2	3.67E-02	2.20E-02	1.39E-02	5.06E-01
Adv-Trained	Yes	FGM	3.37	4.14E-02	3.36E-02	2.26E-02	5.12E-01
Adv-Trained	Yes	PGD	4.46	5.52E-02	4.46E-02	3.91E-02	5.18E-01
Adv-Trained	Yes	Spatial	2.21	2.26E-02	2.21E-02	1.42E-02	5.06E-01
Adv-Trained	Yes	Square	0	0.00E+00	0.00E+00	0.00E+00	4.95E-01
Mixed-Trained	No	None	23.91	2.29E-01	2.39E-01	2.31E-01	6.16E-01
Mixed-Trained	No	FGM	0.98	1.34E-02	9.80E-02	4.80E-03	5.00E-01
Mixed-Trained	No	PGD	1.16	1.89E-02	1.16E-02	9.20E-03	5.01E-01
Mixed-Trained	No	Spatial	18.24	1.88E-01	1.82E-01	1.78E-01	5.84E-01
Mixed-Trained	No	Square	0	0.00E+00	0.00E+00	0.00E+00	4.95E-01
Mixed-Trained	Yes	None	15.91	1.60E-01	1.60E-01	1.52E-01	5.76E-01
Mixed-Trained	Yes	FGM	0.93	1.12E-02	9.30E-03	4.50E-03	5.00E-01
Mixed-Trained	Yes	PGD	1.26	2.07E-02	1.27E-02	1.06E-02	5.01E-01
Mixed-Trained	Yes	Spatial	11.88	1.35E-01	1.19E-01	1.14E-01	5.55E-01
Mixed-Trained	Yes	Square	0	0.00E+00	0.00E+00	0.00E+00	4.95E-01

Table 3—Model performance using accuracy, precision, recall, F-1 score, and AOC-ROC as evaluation metrics.

4.2 McNemar Significance Tests (Defense vs No Defense)

Model	Attack	chi	p	Note
Clean-Trained	None	254.61	0.00E+00	Significant improvement
Clean-Trained	FGM	0.51	4.77E-01	No measurable effect
Clean-Trained	PGD	3.72	5.38E-02	Marginal improvement
Clean-Trained	Spatial	179.74	0.00E+00	Strong improvement
Clean-Trained	Square	0	1.00E+00	No discernable difference
Adv-Trained	None	8.78	3.10E-03	Significant improvement
Adv-Trained	FGM	0.45	5.01E-01	No difference
Adv-Trained	PGD	2.98	8.43E-02	Weak evidence of improvement
Adv-Trained	Spatial	3.09	7.88E-02	Possible effect
Adv-Trained	Square	inf	0.00E+00	Strong effect
Mixed-Trained	None	297.75	0.00E+00	Strong effect
Mixed-Trained	FGM	0.11	7.45E-01	No difference
Mixed-Trained	PGD	0.42	5.16E-01	No difference
Mixed-Trained	Spatial	213.35	0.00E+00	Significant improvement
Mixed-Trained	Square	inf	0.00E+00	Significant improvement

Table 4—Results of the McNemar Chi-Squared Significance test on the accuracy of each model-attack combination examining the significance of the performance differences between having and not having inference defenses.

From the McNemar Chi-Squared test outcomes in Table 4, it can be concluded that inference defenses make significant improvements against the square and spatial attacks under clean and mixed training conditions, whereas only the square attack is mitigated by inference defenses in the adversarial-trained model. Interestingly, the effect of inference defenses on each model varies against the PGD and FGM attacks warranting further investigation.

4.3 Confusion Matrices

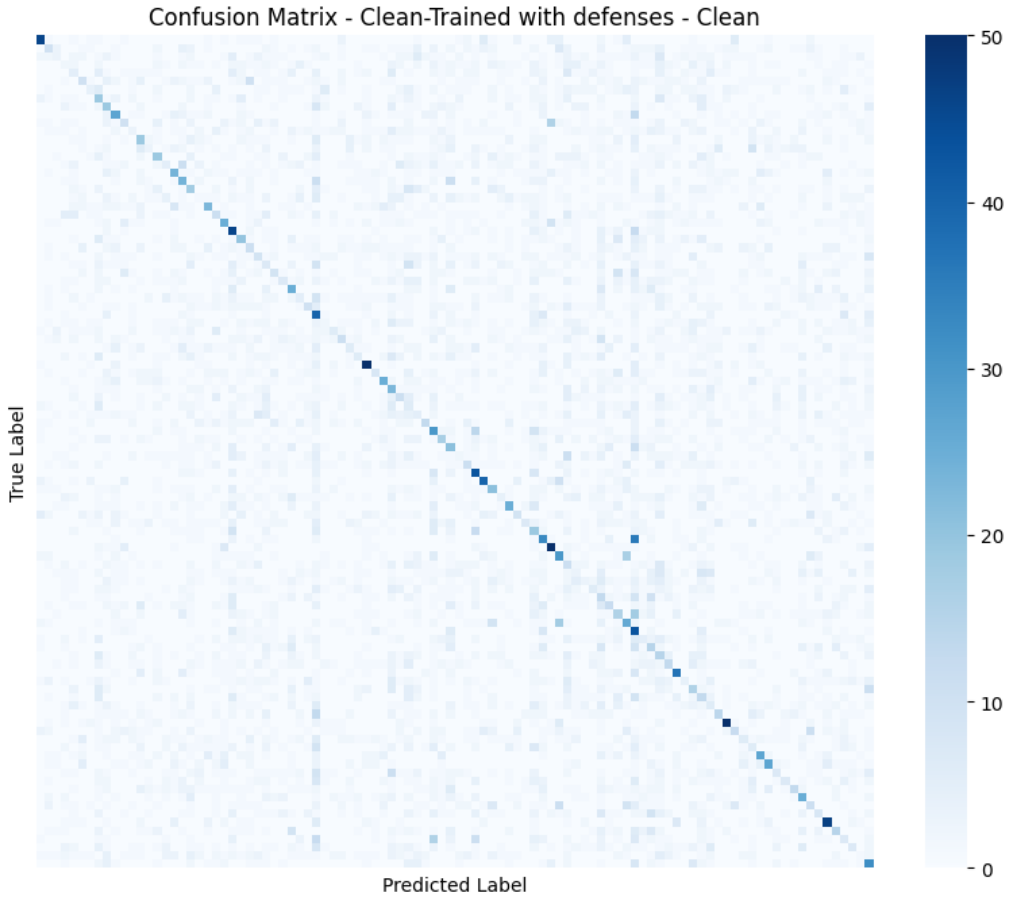

Figure 7—The confusion matrix of the Clean-trained model with inference defenses.

The confusion matrix of the clean-trained model with defenses on clean input (no attack) is shown in Figure 7. The confusion matrix of a clean trained model against FGM without inference defenses (illustrated in Figure 8), demonstrates an exemplary behavior that can also be seen against PGD under the same conditions.

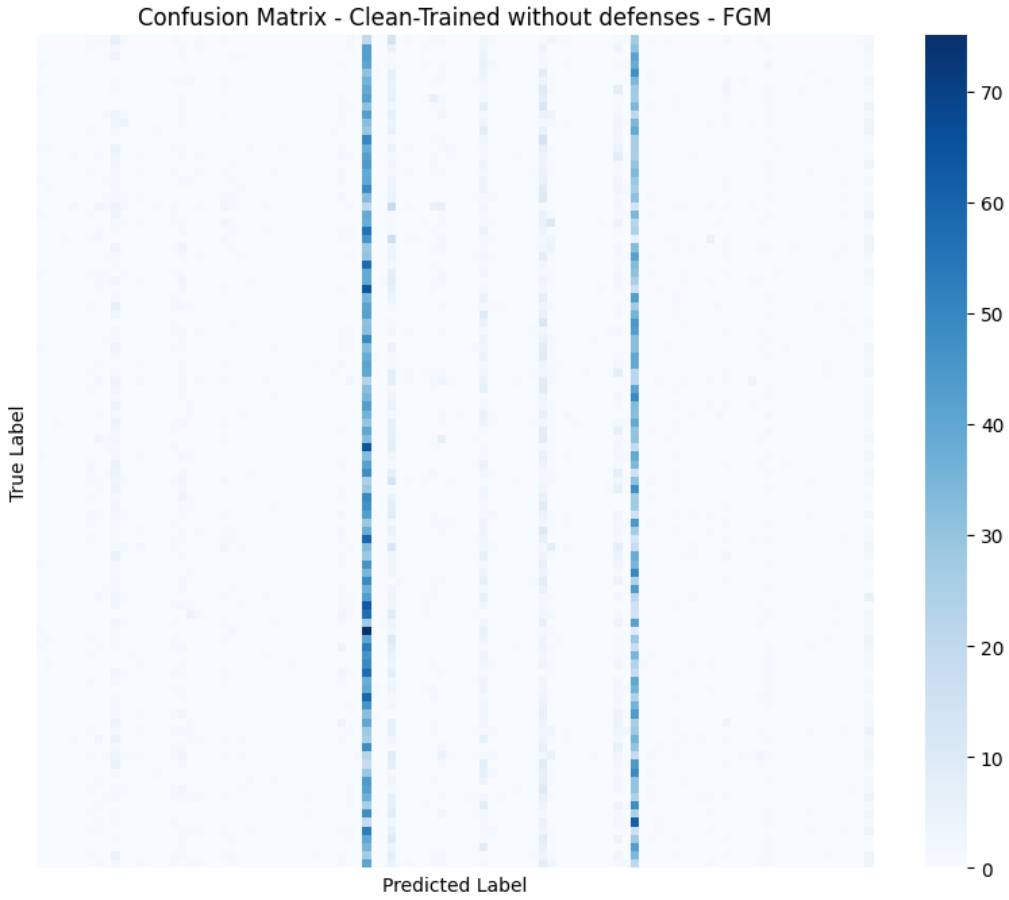

Figure 8—The confusion matrix of a clean-trained model against the FGM attack without using inference defenses.

5 Future Work

1. Study effects of full ablation of training config, defenses, and attack types
2. Explore tandem attacks/defenses and possible interactions, especially with PGD
3. Run experiments on YOLO-cls model with only the final FC layer replaced for CIFAR-100 & freeze the rest of the network
4. Evaluate more attack methods *and* defenses
5. Try other attacks against detection, segmentation, & tracking tasks
6. More exhaustive fine-tuning of classification models, attacks, and defensive strategies

6 References

J. Terven and D. Cordova-Esparza, "A comprehensive review of YOLO architectures in computer vision: From YOLOv1 to YOLOv8 and YOLO-NAS," *Mach. Learn. Knowl. Extr.*, vol. 5, no. 4, pp. 1680-1716, 2023.

M. Hussain, "YOLOv1 to v8: Unveiling each variant—A comprehensive review of YOLO," *IEEE Access*, vol. 12, pp. 42816-42833, Mar. 2024.

I. J. Goodfellow, J. Shlens, and C. Szegedy, “Explaining and harnessing adversarial examples,” *arXiv preprint* arXiv:1412.6572, Dec. 2014.

Goodfellow, I. et al. "Explaining and Harnessing Adversarial Examples." ICLR 2015.

Madry, A. et al. "Towards Deep Learning Models Resistant to Adversarial Attacks." ICLR 2018.

Andriushchenko, M. et al. "Square Attack: A Query-Efficient Black-Box Adversarial Attack." ECCV 2020.

C. Xiao *et al.*, “Spatially transformed adversarial examples,” *arXiv preprint* arXiv:1801.02612, Jan. 2018.

Cohen, J. et al. "Certified Adversarial Robustness via Randomized Smoothing." ICML 2019.

Ultralytics. "YOLOv8 Documentation." 2024.

D. Tsipras, S. Santurkar, L. Engstrom, A. Turner, and A. Madry, “Robustness may be at odds with accuracy,” *arXiv preprint* arXiv:1805.12152, Sep. 2019, v5.

N. Das *et al.*, “SHIELD: Fast, practical defense and vaccination for deep learning using JPEG compression,” *arXiv preprint* arXiv:1802.06816, Feb. 2018.

W. Xu, D. Evans, and Y. Qi, “Feature squeezing: Detecting adversarial examples in deep neural networks,” *arXiv preprint* arXiv:1704.01155, Dec. 2017; to appear in *NDSS 2018*.