AI TEAM TRAINING'26

TASK 1

Phase II

Task 1.3 : A Memo from 221B Baker Street

My dear investigator,

A shadow has fallen over London. My arch-nemesis, Professor Moriarty—that Napoleon of crime—has evolved. He has moved beyond the physical world and now operates in the digital ether, striking at the very heart of our modern security: Artificial Intelligence.

Scotland Yard has deployed a network of state-of-the-art surveillance cameras, each powered by a sophisticated Convolutional Neural Network. Yet, Moriarty's agents slip past them like ghosts. In one instance, a known henchman was recorded walking past a camera, but the system's log registered only 'a potted plant.' In another, a getaway vehicle was classified as 'a library bookshelf.'

Moriarty has developed what I can only describe as a ***"Digital Cloak"****—an imperceptible visual distortion that renders his agents invisible to our AI sentinels. He is exploiting a fundamental flaw in their perception, a flaw I cannot see with my own eyes. My methods are insufficient here; I require your unique expertise.*

Your mission is to step into the role of my AI specialist unit. You must first understand Scotland Yard's technology, then replicate Moriarty's diabolical methods, and finally, develop a countermeasure to harden our defenses. You are to be my digital magnifying glass, revealing the truth hidden within the machine's mind.

The game, as they say, is afoot.

Yours,
S. Holmes

Your Objective

To build and analyze a pipeline in PyTorch that demonstrates how different adversarial attacks can fool various CNN architectures, and to use multiple explainability techniques to diagnose the mechanism of failure. Finally, you will implement and evaluate a defensive technique to improve model robustness.

Phase 1: Model Preparation & Fine-Tuning

- **Objective:** Prepare two high-performance image classification models using transfer learning.
- **Dataset:** Use the Caltech 101 dataset, accessed via torchvision.datasets. Use DataLoader for efficient batching and processing.
- **Architectures:** Select and fine-tune **at least ONE** pre-trained architecture from torchvision.models. Recommended models for comparison are ResNet-34 and MobileNetV2.
- **Procedure:** For each model, freeze all the pre-trained convolutional layers **EXCEPT THE LAST ONE** and replace the final fully-connected layer to match the number of classes in the Caltech 101 dataset. Fine-tune the model until it achieves a strong baseline accuracy on the validation set.

Phase 2: Adversarial Attack Implementation

- **Objective:** Implement methods to generate adversarial examples that fool the models from Phase 1.
- **Techniques:** Implement **at least ONE** adversarial attack method. Here are some examples:
    1. **Fast Gradient Sign Method (FGSM)**
    2. **Jacobian-based Saliency Map Attack (JSMA)**

You can research and implement methods different from these as well. These are just examples.

- **Procedure:** Create functions that apply these attacks to input images. Experiment with the attack strength (epsilon) to find the minimum perturbation required to cause a consistent misclassification.

Phase 3: Model Explainability Implementation

- **Objective:** Implement diagnostic tools to visualize and understand model predictions.
- **Techniques:** Implement **at least TWO** model explainability (XAI) techniques. Here are some examples
    1. **Saliency Maps (Vanilla Gradients)**
    2. **Grad-CAM**

You can research and implement methods different from these as well. These are just examples.

- **Procedure:** Develop a visualization pipeline that can generate these explainability maps for any given image and model prediction, producing a visual output (e.g., a heatmap overlay).

Phase 4: Forensic Analysis

- **Objective:** Synthesize the tools from the previous phases to conduct a detailed analysis of model vulnerabilities.
- **Procedure:** For several interesting examples, create a full "forensic analysis," which includes:
    1. The model's original, correct prediction on a clean image, along with its corresponding XAI visualization.
    2. The adversarially attacked version of the image, the model's new, incorrect prediction, and a visualization of the adversarial noise.
    3. The XAI visualization for the *incorrect* prediction, used to diagnose *why* the model failed.
- **Analysis:** In your final report, you must compare the results across different architectures and attack/explainability methods.

Phase 5: The Countermeasure Protocol (Model Hardening)

- **Objective:** Research, implement, and evaluate a defensive technique to make a model more resilient to the attacks from Phase 2.
- **Techniques:** Choose and implement **at least ONE** defense strategy. You are encouraged to research the trade-offs before selecting one. **You can research and implement methods different from these as well. These are just examples.**
    - **Adversarial Training:** The most direct and basic defense. Modify your training loop to generate adversarial examples for each batch and train the model to correctly classify *both* the clean and attacked images. This teaches the model to recognize its own weaknesses.

- ○ **Defensive Distillation:** A more subtle technique that aims to "smooth" the model's decision surface, making gradients smaller and harder for attackers to exploit. This involves a two-stage "teacher-student" training process as described in the provided research paper.
- ○ **Input Transformation Defenses (Heuristic):** A simpler approach that disrupts the adversarial pattern before it reaches the model. Implement a pre-processing step that applies random transformations (e.g., random resizing + padding, JPEG compression) to the input image at inference time.

- **Evaluation (The Showdown):** Regardless of the method chosen, you must quantitatively prove its effectiveness. Report the accuracy of both your *original* (Phase 1) and *hardened* (Phase 5) models on two versions of the test set:
  - The original, clean test set.
  - An adversarially attacked version of the test set (using your strongest attack).

Deliverables

1. **Code (Jupyter or Python):** A well-structured and documented github repository containing the complete code for all five phases. All model-related code must be written in PyTorch.
2. **Report (PDF or Markdown):** A formal report detailing your investigation. This document is the primary deliverable and must include:
   - **Adversarial Attack Documentation:** A clear explanation of the attack methods you implemented, with code snippets and a comparison of their effectiveness.
   - **Defense Documentation:** An explanation of the defense mechanism you chose to implement, detailing your methodology and any key parameters. This section must be supported by the quantitative "Showdown" results that prove its effectiveness.
   - **Model Explainability Analysis:** The core forensic section. It must contain a rich, visual analysis of your findings, showing a complete set of comparisons. For example:
     - **Saliency/Grad-CAM** maps on a clean image vs. an attacked image.
     - **Saliency/Grad-CAM** maps for the original model vs. the hardened model when faced with the same adversarial attack. (Does the hardened model focus on the correct features even when under attack?)