TML Assignment 2

For this assignment, a **self-supervised learning** approach was chosen over the **semi-supervised MixMatch method**. This decision was based on the nature of the victim model's output: it returns high-dimensional feature vectors rather than low-dimensional class labels, which are typically required for MixMatch-style learning. Since MixMatch is designed to work with soft labels and classification tasks, it is less suitable for direct feature regression tasks such as this one.

The process began with **dataset preparation**. The provided dataset, `ModelStealingPub.pt`, was loaded using a custom `TaskDataset` class. To ensure compatibility with the victim model, image normalization was applied using the specified mean and standard deviation values, aligning the input distribution with what the victim model expects.

Next, the **victim model** was queried. A new API session was initiated to obtain a unique SEED and PORT for communication with the model's endpoint. In this instance, the seed was `36604352` and the port was `9340`. Images were sent in batches of 1000 to the API, with a short delay between requests to avoid blocking. The API returned 1024-dimensional feature representations for each image. These outputs, along with the corresponding image IDs and original images, were saved in a file named `outv1.pickle` for later use.

To **augment the training data**, a data augmentation pipeline was implemented. This included transformations such as color jitter, random horizontal flipping, rotation, and Gaussian blur. The augmented images and their corresponding (reused) feature representations were saved in `augmented_outv1.pickle`. To minimize the number of API queries, only **one representation** was retrieved per original image and reused across all its augmented variants.

Following this, the **stolen model** was trained. The augmented dataset was curated such that only two samples per unique image ID were retained, promoting dataset balance. A ResNet-18 model was used as the base architecture. The first convolutional layer was modified to handle 32×32 RGB images, and the final fully connected layer was adjusted to output 1024-dimensional feature vectors to match the victim model's output. Training was performed using **Mean Squared Error (MSE) loss**, with the goal of minimizing the difference between the stolen model's output and the victim model's representations.

After training, the model was saved as `stolen_model.pth`. For submission, it was exported to ONNX format as `stolen_model.onnx`, with explicitly defined input and output names. The exported model was validated using `onnxruntime` to confirm that it produced the expected output shape.

The final ONNX model was submitted to the evaluation server using the provided API endpoint, along with the required token (`61729223`) and seed.

File Descriptions

- **`outv1.pickle`**: Contains original dataset images and their corresponding feature vectors obtained from the victim model.
- **`augmented_outv1.pickle`**: Contains augmented versions of the original images and their reused representations.
- **`stolen_model.pth`**: The PyTorch model trained to mimic the victim model's feature extractor.
- **`stolen_model.onnx`**: The exported ONNX model ready for submission and evaluation.