Using the Black-Box (Compiled Binaries)

This assignment includes a **compiled "black box" program** that takes an input image, applies **one hidden transformation**, and writes the transformed image back out.

There is one binary for each major platform:

- `black_box_m1` (macOS Apple Silicon)
- `black_box_linux` (Linux x86_64)
- `black_box_windows.exe` (Windows x86_64)

What the program does

- **Input:** an image file (PNG recommended).
- **Output:** a PNG image written to the output path you provide.
- Internally, the program chooses one of the following transformations using a hidden rule, then applies it:
  - invert
  - horizontal flip
  - vertical flip
  - scanlines (zeroes every other row)

Command-line interface (same on all platforms)

The compiled oracle expects **exactly two arguments**:

- `<input_image>`: path to an existing image
- `<output_image>`: path where the result will be saved (PNG)

General form:

- `black_box_<platform> <input_image> <output_image>`

If you run it with the wrong number of arguments, you'll see a usage message like:

- `Usage: <program> <input_image> <output_image>`

macOS (Apple Silicon / M1 / M2 / M3)

From the repository root:

- `./black_box_m1 input.png output.png`

If you get a "Permission denied" error:

- `chmod +x ./black_box_m1` If macOS Gatekeeper blocks execution (quarantine attribute), you have two common options:

1. Finder: right-click the binary → **Open** → confirm.
2. Terminal (advanced):
    - `xattr -dr com.apple.quarantine ./black_box_m1`

Notes:

- This binary is for **Apple Silicon** Macs. If you have an Intel Mac, use the `black_box_orig` version.

Linux (x86_64)

From the repository root:

- `./black_box_linux input.png output.png`

If you get a "Permission denied" error:

- `chmod +x ./black_box_linux`

Windows (x86_64)

Open **PowerShell** (recommended) in the repository folder and run:

- `./black_box_windows.exe input.png output.png`

If Windows SmartScreen warns you, you may need to click **More info → Run anyway**.

Quick sanity check

If you want to verify it's working end-to-end, pick any image (PNG preferred) and run:

- macOS: `./black_box_m1 input_img.png output_img.png`
- Linux: `./black_box_linux input_img.png output_img.png`
- Windows: `./black_box_windows.exe input_img.png output_img.png`

Then open the output image and confirm it looks transformed.

Tips for ML training pipelines

- **Be consistent about image format and size.** For Fashion-MNIST/MNIST, saving/loading **28×28 grayscale PNGs** is a good default.
- The compiled oracle will grayscale inputs automatically; color inputs are allowed.
- If you're calling the binary from Python, you can use `subprocess.run([...], check=True)`.