CISPA HELMHOLTZ CENTER FOR INFORMATION SECURITY

Task: Watermark Detection

Goal

The goal of this task is to determine whether a given image is watermarked or not (clean) using continuous prediction scores.
Each participant must design a detector that outputs a numeric score in the range [0,1], where higher values indicate a higher likelihood of being watermarked. The evaluation then determines performance based on how well these scores separate clean from watermarked images.

Challenge

The dataset contains images that may have been processed using watermarking techniques. The main challenge is to design a robust and general detector capable of identifying watermarks regardless of the underlying embedding scheme.
The evaluation emphasizes maintaining a low false positive rate (FPR) while maximizing the true positive rate (TPR).

Evaluation Metric

Submissions are evaluated using True Positive Rate (TPR) at 1 % False Positive Rate (FPR).
Each model must output a continuous confidence score between 0 and 1 for every test image, indicating how likely it is that the image is watermarked.

Scores ≥ 0.5 → watermarked
Scores < 0.5 → clean

These scores are used to measure how well the model separates watermarked from clean images.
A high TPR means the detector correctly identifies many watermarked images.
The FPR is fixed at 1 %, meaning that only 1 % of clean images are allowed to be falsely marked as watermarked.
The final leaderboard score represents the TPR value achieved at 1 % FPR.
Higher scores indicate better watermark detection performance.

Dataset

A single archive (Dataset.zip) is provided containing three predefined splits:

1. Train Split - 320 images total
    a. 160 labeled “clean”
    b. 160 labeled “watermark” (split evenly across watermark types)
2. Validation Split
    a. 80 labeled “clean”
    b. 80 labeled “watermark”
3. Test Split

a. 2,000 images total

Total images: 2,480

- (also available at HuggingFace or on juelich login nodes at /p/project1/training2557/common/image-attribution)

The test split is used exclusively for leaderboard scoring.Only a portion of 30% contributes to the public leaderboard, while the remaining 70% determines the final private ranking.

Additional Resources
A starter kit is provided to help participants begin quickly.
It includes:

- Code to unzip and load the dataset into PyTorch datasets
- Utility functions for training and evaluation
- Example code to generate and save continuous predictions for submission

Participants can modify or extend the provided code with custom architectures, preprocessing, or features.
Submissions must follow the required file format described below.

Submission Format
The submission file must be named submission.csv and follow the structure below:

```
image_name,score
1.png,0.8321
2.png,0.0478
3.png,0.6214
…
```

Requirements

- Each test image must appear exactly once
- Column names must be exactly image_name and score (case-sensitive)
- Scores must be numeric values between 0 and 1
- No duplicates, missing rows, or invalid entries
- File size must not exceed 10 MB

Submissions that do not meet these requirements will be rejected automatically.

Scoring
All submissions are automatically validated and evaluated on the hidden test set.

- Public leaderboard: based on 30% of test images
- Private leaderboard: final ranking based on the remaining 70%

The leaderboard score reflects the TPR @ 1% FPR achieved by the submission.
A higher score indicates stronger watermark detection performance.

How to Get Started

1. Install Dependencies - pip install torch torchvision pillow numpy pandas requests scikit-learn
2. Place the Dataset - Download and place dataset.zip in the same directory as the starter script.
3. Run the Starter Kit - python task_template.py
   This will: Unzip the dataset, Load the train, validation, and test splits and save sample continuous predictions in submission.csv

Before submitting, ensure that:

- Every test image is listed once
- Column names match exactly (image_name, score)
- All scores are within [0, 1] and contain no missing or invalid values

Submit to Leaderboard

Insert the provided team token into the submission script and upload results using the command below:

```
import pandas as pd
import requests

df = pd.DataFrame({
    "image_name": image_names,
    "score": predictions,
})
df.to_csv("submission.csv", index=False)

response = requests.post(
    "http://34.122.51.94:9000",
    files={"file": open("submission.csv", "rb")},
    headers={"token": "YOUR_TEAM_TOKEN"}
)
print(response.json())
```

Leaderboard

After evaluation, your results can be found in the leaderboard:

- You can access the leaderboard for this task at http://34.122.51.94:80/leaderboard_page. This will help you to compare your solutions with other teams and see where you stand.

- The leaderboard shows the best result per team only. As output to your request, you will get back the score for your current submission. If it is lower than the score saved in the leaderboard, the score will not be updated.

References:

1. Louis Kerner, Michel Meintz, Bihe Zhao, Franziska Boenisch, Adam Dziedzic. “BitMark: Watermarking Bitwise Autoregressive Image Generative Models” https://openreview.net/forum?id=VSir0FzFnP

2. Yuxin Wen, John Kirchenbauer, Jonas Geiping, Tom Goldstein “Tree-Rings Watermarks: Invisible Fingerprints for Diffusion Images” NeurIPS 2023 https://openreview.net/forum?id=Z57JrmubNl
3. Pierre Fernandez, Guillaume Couairon, Hervé Jégou, Matthijs Douze, Teddy Furon “The Stable Signature: Rooting Watermarks in Latent Diffusion Models” ICCV 2023 https://arxiv.org/abs/2303.15435