**Adversarial Neural Cryptography for Enhanced Steganographic Security**

This repository contains the code and resources for the paper titled **"Adversarial Neural Cryptography: Enhancing Steganographic Security and Robustness."** This work introduces a novel adversarial neural network framework for steganography, where an encoder (Alice) embeds secret messages within images, a decoder (Bob) retrieves these messages, and an adversary (Eve) attempts to detect them. The approach is designed to increase the security and robustness of hidden communication.

**Table of Contents**

* [Overview](#overview)
* [Repository Structure](#repository-structure)
* [Installation](#installation)
* [Usage](#usage)
* [Evaluation Metrics](#evaluation-metrics)
* [Results](#results)
* [Citation](#citation)

**Overview**

This project leverages **Adversarial Neural Cryptography (ANC)** to improve the security of message embedding through a continuous adversarial training framework:

* **Encoder (Alice)** hides messages within images.
* **Decoder (Bob)** accurately retrieves the messages from encoded images.
* **Adversary (Eve)** attempts to detect the presence of hidden messages, forcing Alice and Bob to improve their robustness.

Key features of the implementation include:

* Integration of Mean Squared Error, Structural Similarity Index Measure (SSIM), and Perceptual Loss functions.
* Experiments conducted on CIFAR-10 to showcase the model's effectiveness over 50 epochs.
* Code designed to replicate all results presented in the paper.

**Repository Structure**

* src/: Contains the main code for training and testing.
  + train.py: Script to train the adversarial network with Alice, Bob, and Eve.
  + test.py: Script to evaluate the trained model.
  + model.py: Defines the neural network architectures for Alice, Bob, and Eve.
  + data\_loader.py: Handles loading and preprocessing of datasets.
  + utils.py: Utility functions, including loss calculations and metrics.
* data/: Contains sample datasets (or instructions to download CIFAR-10).
* docs/: Documentation, diagrams, and supplementary materials.
  + block\_diagram.png: Diagram of the execution process.
* README.md: This file, describing the repository and usage.
* requirements.txt: List of dependencies required for the project.

**Installation**

1. **Clone the repository:**

bash

Copy code

git clone https://github.com/YourUsername/Adversarial-Neural-Cryptography.git

cd Adversarial-Neural-Cryptography

1. **Create a virtual environment (optional but recommended):**

bash

Copy code

python3 -m venv env

source env/bin/activate # For Linux/Mac

env\Scripts\activate # For Windows

1. **Install dependencies:**

bash

Copy code

pip install -r requirements.txt

1. **Download CIFAR-10 Dataset** (if not included):
   * The data\_loader.py script automatically downloads CIFAR-10 if not present. Alternatively, you can manually download CIFAR-10 and place it in the data/ folder.

**Usage**

1. **Training the Model:** Run the training script with default settings:

bash

Copy code

python src/train.py

* + Modify training parameters (e.g., epochs, batch size) in train.py or via command-line arguments as needed.

1. **Testing the Model:** Once trained, you can test the model with:

bash

Copy code

python src/test.py

* + This will generate results, including message recovery rates and detection resistance metrics.

1. **Configuration:**
   * Parameters (e.g., learning rate, number of epochs, batch size) can be adjusted in train.py.
   * Modify model.py to experiment with the neural network architectures.

**Evaluation Metrics**

The following metrics are used to evaluate the model's performance:

* **Message Recovery Accuracy (MSE)**: Mean Squared Error to assess how accurately Bob recovers the original message.
* **Detection Resistance (Binary Cross-Entropy Loss)**: Measures Eve’s ability to detect the hidden message.
* **Image Quality (SSIM and Perceptual Loss)**: Evaluates the visual similarity between the cover image and the encoded image to ensure imperceptibility.

**Results**

Our experiments on CIFAR-10 demonstrate the following:

* Significant improvements in message recovery and resistance to detection attacks.
* A robust ANC-based steganographic system that achieves high security and robustness.

For detailed results and comparisons, refer to our paper.

**Citation**

If you find this code useful in your research, please cite our paper:

bibtex

Copy code

@article{YourPaperReference,

title={Adversarial Neural Cryptography: Enhancing Steganographic Security and Robustness},

author={Basil Hanafi, Mohammad Ubaidullah Bokhari, Imran Khan},

journal={The Visual Computer},

year={2024},

doi={<https://doi.org/10.5281/zenodo.14046144>}

}

**License**

This project is licensed under the MIT License. See the <LICENSE> file for details.