**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 an innovative adversarial neural cryptography (ANC) framework 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 enhance the security and robustness of hidden communication.

Table of Contents

* Overview
* Repository Structure
* Installation
* Usage
* Evaluation Metrics
* Results
* Citation
* License

**Overview**

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

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

Key features of the implementation include:

* Integration of Mean Squared Error, Structural Similarity Index Measure (SSIM), and Perceptual Loss functions.
* Experiments on the CIFAR-10 dataset to demonstrate model 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/basilhanafics/ANCforSteganography.git

cd ANCforSteganography

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 will automatically download CIFAR-10 if not present. Alternatively, 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: After training, 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:
   * Adjust parameters (e.g., learning rate, epochs) directly in train.py.
   * Experiment with the neural network architectures in model.py for custom implementations.

**Evaluation Metrics**

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

* Message Recovery Accuracy (MSE): Mean Squared Error to assess Bob's accuracy in message recovery.
* Detection Resistance (Binary Cross-Entropy Loss): Measures Eve’s ability to detect hidden messages.
* Image Quality (SSIM and Perceptual Loss): Assesses the similarity between cover and encoded images to ensure imperceptibility.

**Results**

Our experiments on CIFAR-10 demonstrate:

* Improved message recovery and resistance to detection attacks.
* High security and robustness in ANC-based steganography.

Example results, including loss curves and sample images (cover, encoded, recovered), are provided in the docs/ folder.

**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},

journal={The Visual Computer},

year={2024},

doi={10.5281/zenodo.14059348}

}

**License**

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