DeepSteg - A Deep Learning Approach for steganographic detection

EE6310 | Image and Video Processing

Haritha(AI20BTECH11010) Adhvik (AI20BTECH11015)
Arun(AI20BTECH11019) Jaswanth(EE20BTECH11025)

IIT Hyderabad

May 3, 2023

भारतीय प्रौद्योगिकी संस्थान हैदराबाद
Indian Institute of Technology Hyderabad

Contents

1. Introduction
2. Problem Statement
   - Steganography and Steganalysis
3. Literature review
4. Approaches Explored
5. Implementation
6. Results
7. Learnings from the project
8. Future work

Introduction

Introduction

- Steganography is a technique of hiding secret information within a non-secret file. It has become a popular tool for ensuring data security.
- Steganalysis is the process of detecting and decoding hidden messages within a file. Deep learning has potential in steganography.
- This project intends to explore various deep learning methods for detecting and decoding steg Images.

Problem Statement

Steganography and Steganalysis

- Steganography is the intersection of cryptography, information theory, and machine learning Domains.
- It can be broadly studied under two categories: Spatial Domain and Frequency Domain.
- In this project we are focusing on JPEG Image steganography and steganalysis (using DL)

Literature review

Literature Review

- We focused our further literature review on Steganographic methods and DL based steganalysis.
- Although traditional steganalysis methods are performing well some require prior knowledge of the algorithm used for encoding for better accuracy.
- Why DL based techniques?Link
  - Fundamentally, the detection of modern content-adaptive steganography is equivalent to detecting noise-like signals shaped by the content itself.
  - It is thus not surprising that CNNs trained on computer vision tasks are a good starting point for transfer learning in steganalysis, as well as the closely related field of digital forensics.

Approaches Explored

Steganography

1. LSB - [paper] [code]
2. JUNIWARD - [paper] [code]
3. JMiPOD - [ppt] [code]
4. UERD - [ppt] [code]

Steganalysis

1. Deep Learning method for JUNIWARD Detection - [paper]
2. kaggle challenge - [link]
3. [Basic Understanding]
4. [EfficientNet]
5. SRNET implementation [SRNET]
6. [First Place Solution]
7. [Ensemble models]

Implementation

Steganalysis Model

- We have used an Efficientnet-B0 and B2 pretrained model and fine tuned it on the JUNIWARD encoded data from ALASKA2 dataset.
- We used cross-entropy loss and the AdamW optimizer with 10^{-4} weight decay for 60 epochs using a learning rate scheduler with a start LR of 0.001 and end LR of 2×10^{-5}. We used a minimum batch size of 16, which was increased for smaller architectures to speed up training.
- After training, we chose the best checkpoint based on the wAUC metric on the validation set. (With the references we referenced, we expect our model to perform with a score of 0.8.)

Steganographic model

- We took the code from GitHub, a library code (hstego).
- The heuristic design of distortion function is the core of the algorithm. Distortion function is used to evaluate the effect of modification of Image.
- An adaptive steganography algorithm tends to embed message into textured and noisy region of the image which is not easily modellable in any direction.
- The distortion function of J-UNIWARD is constructed by quantifying this with the outputs of three directional filters

Steganographic model

- The J-UNIWARD distortion function is the sum of relative changes of all wavelet coefficients between cover and stego images:

$$D(\mathrm{X},\mathrm{Y}) \triangleq \sum_{k=1}^{3}\sum_{u=1}^{n}\sum_{v=1}^{m} \frac{\left|W_{uv}^{(k)}(\mathrm{X}) - W_{uv}^{(k)}(\mathrm{Y})\right|}{\sigma + \left|W_{uv}^{(k)}(\mathrm{X})\right|}$$

  where $W_{uv}^{(k)}(\mathrm{X})$ and $W_{uv}^{(k)}(\mathrm{Y})$ are uv th wavelet coefficients in k th subband of the first decomposition level.

- For JPEG images, the distortion between quantized DCT coefficients of X and Y is computed by spatial images $J^{-1}(\mathrm{X})$ and $J^{-1}(\mathrm{Y})$ decompressed from JPEG files:

$$D(\mathrm{X},\mathrm{Y}) \triangleq D\left(J^{-1}(\mathrm{X}), J^{-1}(\mathrm{Y})\right)$$

Syndrome-trellis codes (STC)

- When the embedding distortion of each pixel in the cover is obtained, the sender can use syndrome coding to embed message m while minimizing the average distortion:

$$\mathrm{Emb}(\mathrm{X}, \mathrm{m}) = \arg \min_{P(\mathrm{Y}) \in C(m)} D(\mathrm{X}, \mathrm{Y})$$

$$\mathrm{Ext}(\mathrm{Y}) = P(\mathrm{Y}) H^T = \mathrm{m}$$

where $P(\mathrm{Y})$ represents the LSB sequence of stego, $C(\mathrm{m}) = \left\{ \mathrm{z} \in \{0,1\}^n \mid \mathrm{z} H^T = \mathrm{m} \right\}$ is the coset corresponding to syndrome m, and $H^T \in \{0,1\}^{n \times m}$ is a parity-check matrix of $C(m)$, which is constructed by placing a small submatrix $\hat{H}$ of size $h \times \omega (\omega = m/n)$ along the main diagonal. Besides, the width of $\hat{H}$ is dictated by the desired ration of ω, which coincides with the relative payload. - [paper]

App implementation

1. Our app runs on a flask server.
2. We have used pickel file generated from our trained neural network to implement the steganalysis in our app.
3. And We have referenced this code and implemented the JUNIWARD steganography algorithm for the steganography.

Results

Performance of our model

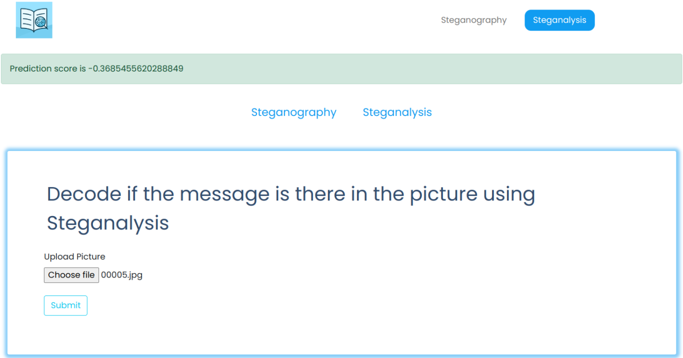
1. We expected a score of around 0.8 (from code references) when submitted in kaggle but it gave around 0.6 as we trained our model with less data.
2. We have got the following results for our best trained model with 20% of the data:

$$Acc : 0.5978571428571429$$
$$Score : 0.583808712927487$$

Performance of our app

Figure: cover image

Steganalysis of Cover image



Prediction score is -0.3685455620288849

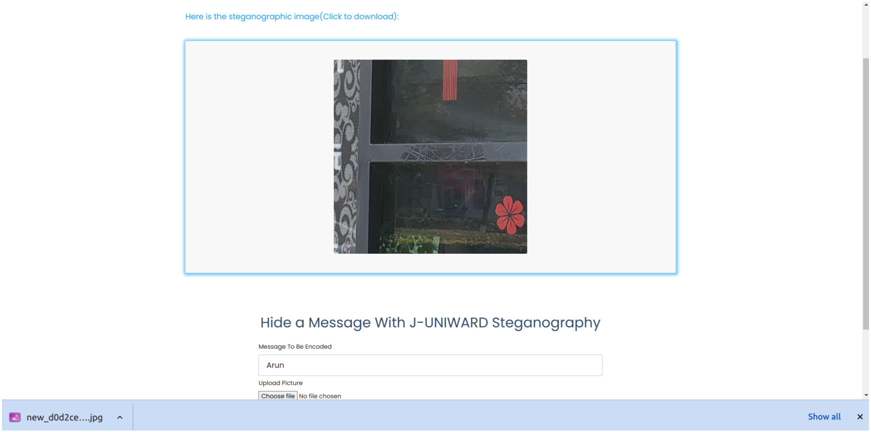
Decode if the message is there in the picture using Steganalysis

Upload Picture

Choose file 00005.jpg

Figure: Sent the cover image to our app

Steganography



Here is the steganographic image(Click to download):

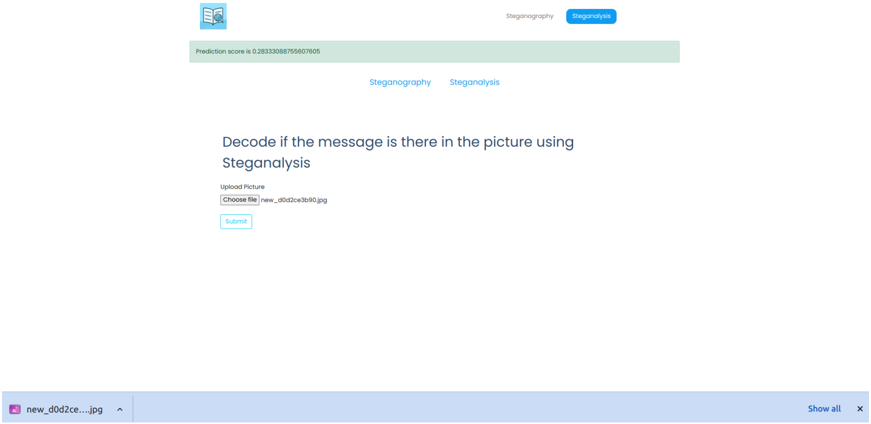
Hide a Message With J-UNIWARD Steganography

Message To Be Encoded: Arun

Upload Picture: Choose file No file chosen

Figure: Sent the cover image to our app for juniward steganography

Steganalysis



Prediction score is 0.28333088755607605

Decode if the message is there in the picture using Steganalysis

Upload Picture

Choose file new_d0d2ce3b90.jpg

Figure: Sent the stego image to our app for steganalysis

Learnings from the project

Learnings

- We have learned about the efficientNet, SRNet and how it can also be used in other domains.
- We have learnt about some LSB steganography
- We have learnt how DCT coefficients can be used to hide information(JUNIWARD, JMiPOD, UERD) steganography.
- We have also seen some uses of JPEG toolbox which is used for handling JPEG images.
- We have explored the STCs how they can be used in images embedding.
- We also learned models trained on YCbcr spatial domain can be used for steganalysis.

Some of our (interesting) Findings

- J-UNIWARD used the daubechies 8-tap wavelet filter bank in its implementation which was discussed in class.
- JMiPOD steganography uses Wiener filter for noise reduction which was discussed in class.
- Even though, steganalysis task is fundamentally different from the main objective of computer vision (object classification) we can use CNN's to perform the task.
- DCT sizes of 2×2, 3×3, 4×4, 5×5, and 8×8 have been tested the best results are obtained with size 4×4.

Future work

Future Work

1. Improving this model and We want to try the ensemble method using both SRNet and EfficientNet.
2. We want to explore more about the first ranker method of solving this problem using seresnet.
3. We can try to implement this paper in the app which discusses about the **decoding** the stego message for plain text embeddings. [paper]
4. We can try improving the J-UNIWARD speed using FS-UNIWARD - [paper]

Contributions

- **AI20BTECH11010:** Traditional steganographic part of steganography and Steganography Code for JUNIWARD
- **AI20BTECH11015:** Literature review of DL based steganography and steganalysis and Fine-tuning the pre-trained model and helped in integrating the trained model to our web app.
- **AI20BTECH11019:** Further Literature review of the DL based steganalysis.Website coding and Integration of the above both(steganography and steganalysis) codes in the website.
- **EE20BTECH11025:** ALASKA2 dataset, Kaggle codes exploration. Steganography Code for JUNIWARD.

Thank you!