Hiding in Plain Strokes: Handwriting and Applications to Steganography

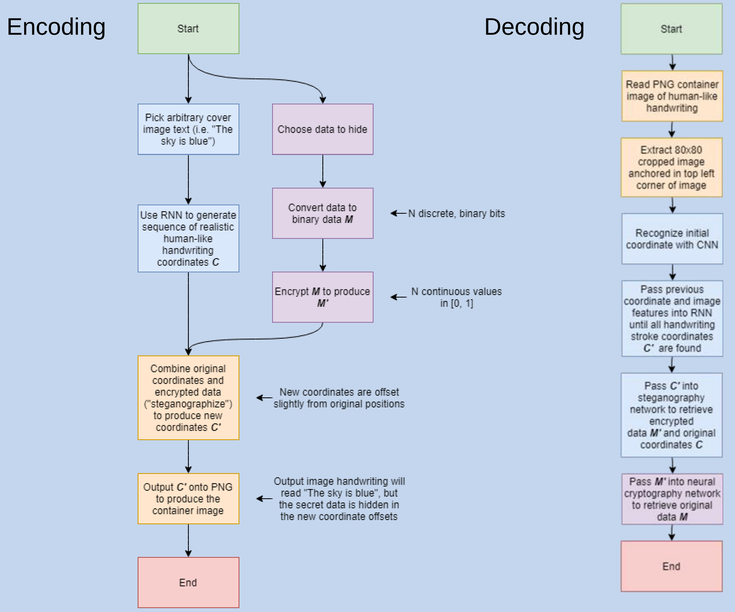
James Hahn[1,2] Adriana Kovashka[1]
[1]University of Pittsburgh [2]Georgia Tech

Motivation

- Steganography is a way to hide a message in plain sight, but existing methods primarily consider dense covers and messages (saturated images)
- Thus it is hard for an adversary to visually detect the encoded message, and easy to automatically extract an approximately correct message once detected
- We consider a complementary type of cover which is sparse, i.e. most of the image is white, with black pixels for handwriting strokes
- Handwritten text is easy to generate hence many covers can be generated; it is natural to share/show handwritten text without raising suspicion (e.g. signatures)

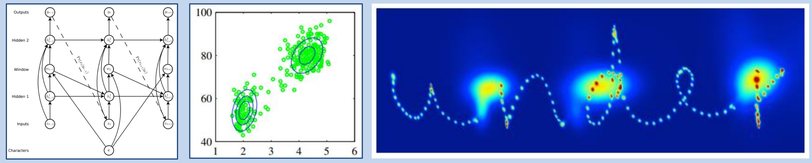
Overview

- We develop a framework for hiding a message for privacy
- We first encrypt the message, then embed it in handwritten text
- We show we are robust to adversaries trying to break the crypto code, that our stego image looks similar to the original cover, and the message can be successfully extracted by the intended recipient



Handwriting generation

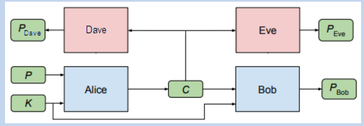
- Graves, "Generating sequences with recurrent neural networks"
- Uses the IAM Online Handwriting Database
- Input: previously predicted coordinate, output: parameter for next coordinate's probability distribution (Mixture of Gaussians)



- Results, with control of style (left) and bias (middle, right):

Message encryption

- Abadi & Andersen, "Learning to protect communications w/ adversarial neural crypto"
- Alice (sender) and Bob (recipient) want Bob to have decryption error of 0% while forcing Eve to be 50% (random guess)
- Eve wants a decryption, or reconstruction, error of 0%
- For robustness and generalizability, we add another adversary, Dave, to learn alongside Eve; Dave's error is not included in Alice and Bob's error calculation



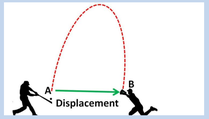
$$L_{Eve} = d(P, Eve(C))$$

$$L_{Bob} = d(P, Bob(C, K))$$

$$L_{Alice} = L_{Bob} - L_{Eve}$$

Steganography

- Train a network that reads in cover pixels and adds offset based on message
- Modification rather than addition to the cover, makes extraction challenging
- Floating point coordinates can be achieved by adding surrounding grey values



Evaluation

Cryptography

	Alice/Bob	Eve	Dave
Loss (bits)	1.7569	62.8310	65.5122
Loss (%)	1.3726%	49.0867%	51.1814%

Steganography

	Cover	Decoding	Total
Loss (bits)	0.1770	0.0029	0.1800

Qualitative results

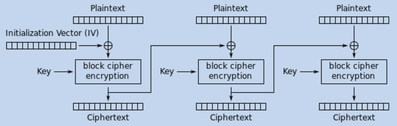
Original cover image

Container image with secret message
"The password is 123"

Container image with secret message
"Meet me on the east side at 11:03pm."

Conclusions, limitations, future work

- Our method is generalizable
  - A variety of handwriting styles can be used
  - Cover image doesn't have to be square
  - Can be applied to non-handwriting domains, e.g. modify attributes
  - Works for image and non-image data
- Evaluation is still incomplete
  - Test detection by adversary for steganographic image
  - Test robustness to adversaries of the complete pipeline (crypto+stego)
- The cover loss is a proxy for achieving a realistic image
  - But an alternative can be obtained through a GAN loss
- Train full system end-to-end
- Chunking is problematic, instead could use CBC (Ehrsam et al., "Message verification and transmission error detection by block chaining")



Cipher Block Chaining (CBC) mode encryption