A Wild Manhunt for Stego Images Created by Mobile Apps

Li Lin, Wenhao Chen, Stephanie Reinders, Min Wu*, Yong Guan, and Jennifer Newman

Iowa State University

*University of Maryland

This work was partially funded by the Center for Statistics and Applications in Forensic Evidence (CSAFE) through Cooperative Agreement #70NANB15H176 between NIST and Iowa State University, which includes activities carried out at Carnegie Mellon University, University of California Irvine, and University of Virginia.

Background

- As mobile Internet and telecommunication technology develops at high speed, the digital image forensics academic community is facing a growing challenge.
- Mobile applications (Apps) allow a user to easily edit/process an image for a variety of purposes.
- Thanks to the improved cameras and editing apps on smartphones, the volume of images presented to digital image forensic practitioners increases every day.
- Unfortunately, terrorists, spies and child pornography predators are also taking the advantage of the mobile app ecosystem to exchange illegal files and photos.

Research funded by the Center for Statistics and Applications in Forensic Evidence (CSAFE) - forensicstats.org

Steganography Apps on Google Play

App Name	Installs	Open Source	Output Format	Image Resizing	Payload Pre-processing			Embedding Technique
					Encryption	Signature Strings	Length Data	
PixelKnot	100,000+	Yes	JPG	Downsampling	Yes	No	Yes	F5
Steganography Master	10,000+	No	PNG	No	No	Yes	No	1's digit replacement
Steganography_M	10,000+	No	PNG	No	No	Yes	No	LSB replacement
DaVinci Secret Image	5,000+	No	PNG	User specified	No	Yes	Yes	Alpha channel encoding
Steganography_T	5,000+	No	PNG	No	No	No	Yes	LSB replacement
Stegais	1,000+	No	JPG	Downsampling	No	No	Yes	Unknown
PocketStego	1,000+	No	PNG	Downsampling	No	Yes	No	LSB
MobiStego	1,000+	Yes	PNG	Downsampling	Yes	Yes	No	RGB channels LS2B
NiaStego	1,000+	No	PNG	Upsampling	Yes	Yes	No	RGB channels LSB
Passlok	1,000+	Yes	JPG	No	Yes	Yes	No	Non-shrinkage F5

Research funded by the Center for Statistics and Applications in Forensic Evidence (CSAFE) - forensicstats.org

Steganography

- Steganography embeds data into an object, so that even the existence of the secret message cannot be discovered by visual observation.

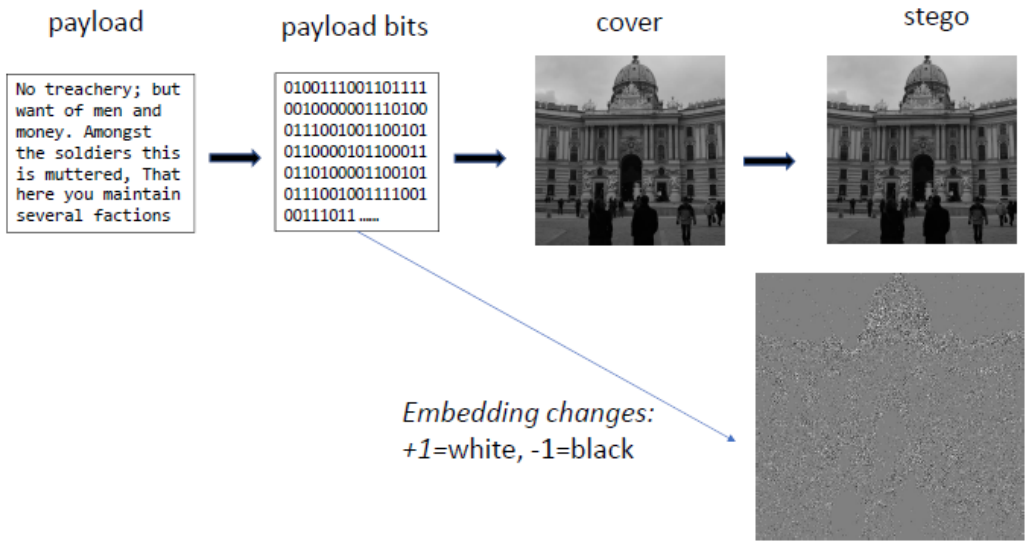


Figure: An example of steganography embedding in the spatial domain by embedding algorithm S-Uniward

Research funded by the Center for Statistics and Applications in Forensic Evidence (CSAFE) - forensicstats.org

Steganography

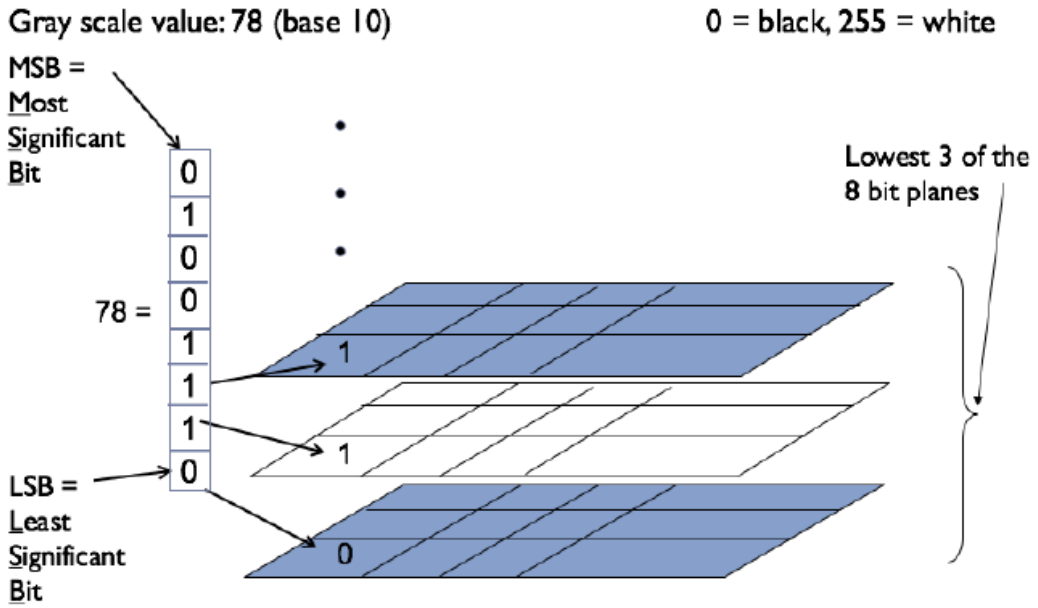
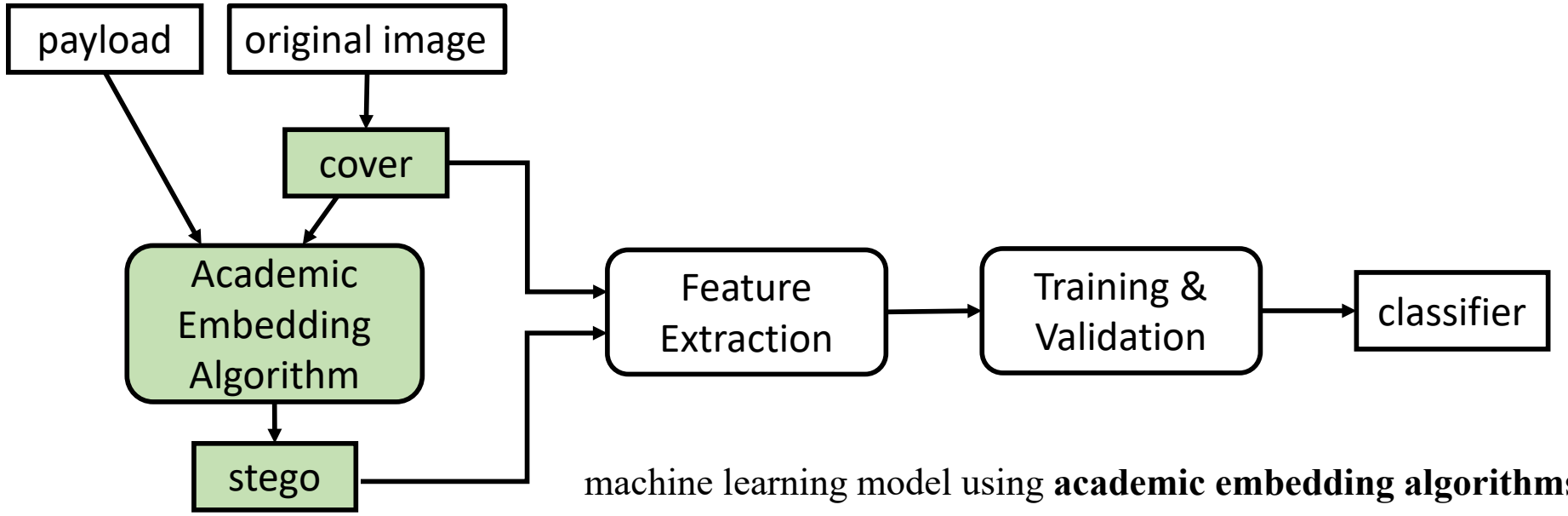


Figure: An example of the Least Significant Bit in the spatial domain

Research funded by the Center for Statistics and Applications in Forensic Evidence (CSAFE) - forensicstats.org

Academic Steganalysis

- The forensic process to detect steganography
- Machine learning models are used extensively in academics

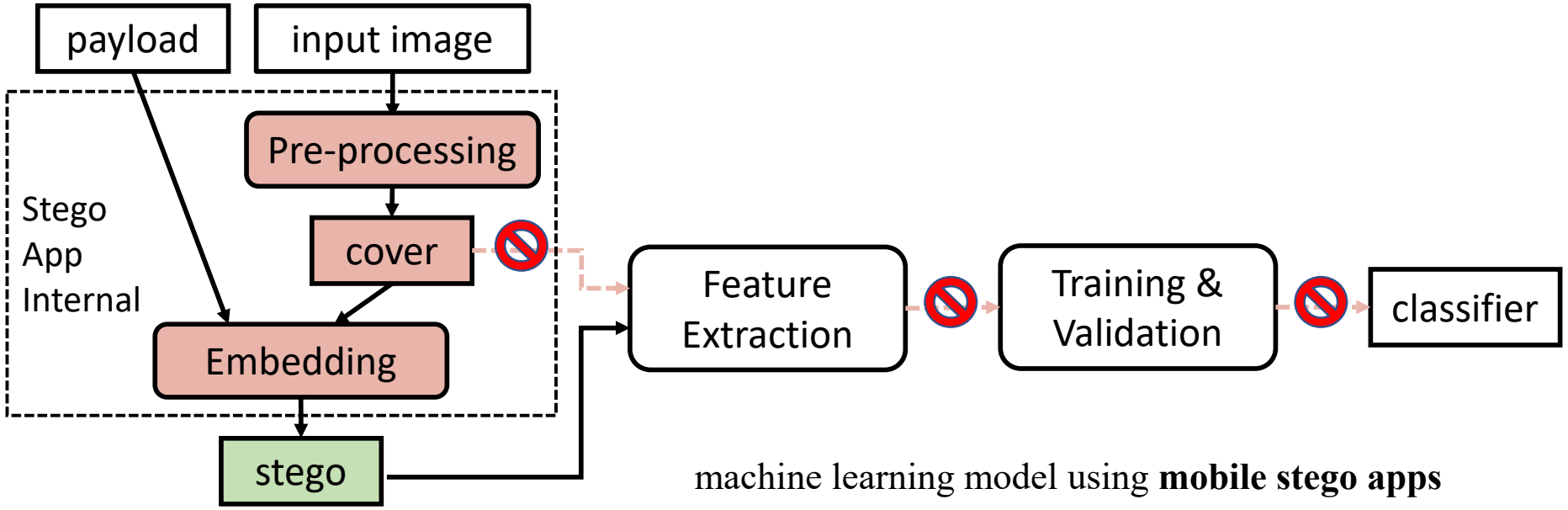


machine learning model using **academic embedding algorithms**

Research funded by the Center for Statistics and Applications in Forensic Evidence (CSAFE) - forensicstats.org

Challenges of Steganalysis for stego apps

- Unknown components when applied to mobile stego apps
  - No access to cover image
  - No control over the embedding rate (% of image data used for embedding)



machine learning model using **mobile stego apps**

Research funded by the Center for Statistics and Applications in Forensic Evidence (CSAFE) - forensicstats.org

Our Contributions

- Construction of a heavily-provenanced digital image reference database, StegoAppDB, that simulates digital evidence and provides data for
  - Testing current steg detection tools – Stego Hunt, DC3 StegDetect
  - Developing tools that detect steg images created from mobile apps

- **StegoAppDB**: A mobile stego image dataset for steganalysis
  - https://forensicstats.org/stegoappdb/

- Detection of stego images from Android apps
  - Detection using machine-learning detection
  - Detection using signature-based methods

Research funded by the Center for Statistics and Applications in Forensic Evidence (CSAFE) - forensicstats.org

StegoAppDB

Device Model	# Devices	ISO Range	Exposure Time Range	# Scenes	# Original Images	# Cropped Images	# Covers	# Stegos
Google Pixel 1	4	107 ~3735	1/120 ~1/10	404	8080	8080	36360	181800
Google Pixel 2	4	67 ~3155	1/252 ~1/12	404	8080	8080	36360	181800
Samsung Galaxy S8	2	57 ~6846	1/120 ~1/12	200	4000	4000	18000	90000
OnePlus 5	2	100 ~3000	1/10830 ~1/15	200	4000	4000	18000	90000
iPhone 6s	4	40 ~1600	1/60 ~1/3	404	8080	8080	8080	40400
iPhone 6s Plus	2	25 ~1250	1/66 ~1/3	213	4260	4260	4260	21300
iPhone 7	4	25 ~1000	1/60 ~1/3	404	8080	8080	8080	40400
iPhone 7 Plus	2	25 ~1000	1/80 ~1/3	202	4040	4040	4040	20200
iPhone 8	2	32 ~1600	1/60 ~1/3	202	4040	4040	4040	20200
iPhone X	2	20 ~1600	1/62 ~1/3	203	4060	4060	4060	20300
total	28	20 ~6846	1/10830 ~1/3	2836	56720	56720	141280	706400
total images								961120

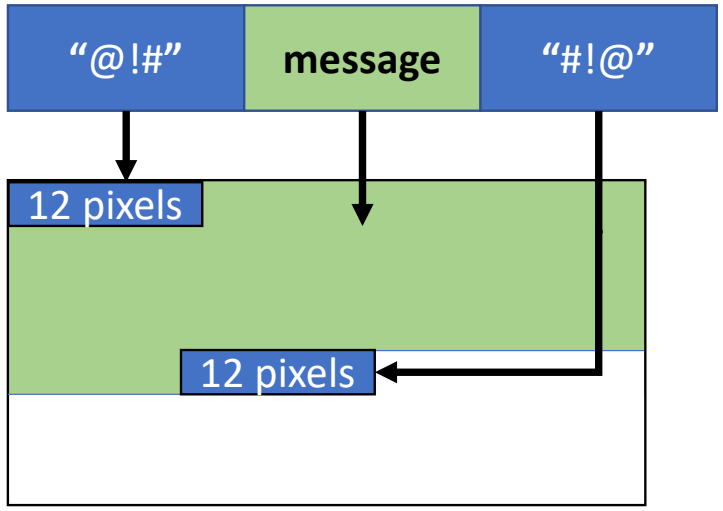
Research funded by the Center for Statistics and Applications in Forensic Evidence (CSAFE) - forensicstats.org

Signature-based Steg Detection

- Signature definition:
  - A fixed bit string pattern extractable from the stego image
  - Constant strings embedded into fixed locations, to demarcate the message
- Example for the app “MobiStego”

Payload composition:

Stego Image:



Research funded by the Center for Statistics and Applications in Forensic Evidence (CSAFE) - forensicstats.org

Signature-based Steg Detection

- Four stego apps contain embedding signatures

Stego App	Payload Composition (constant strings + user input)
Steganography Master	constant (102 bits) | password | constant (24 bits) | message | constant (88 bits)
Da Vinci Secret Image	constant (32 bits) | length (32 bits) | password | length (32 bits) | message
MobiStego	constant (24 bits) | encrypted message | constant (24 bits)
PocketStego	message | constant (8 bits)

Research funded by the Center for Statistics and Applications in Forensic Evidence (CSAFE) - forensicstats.org

Signature-based Steg Detection Results

- Results of signature-based detection on 202,080 images

Stego App	Test Images	Image Count	Accuracy
Steganography Master	SM Stego Images	42,100	100%
	Other Images	159,980	100%
DaVinci Secret Image	DV Stego Images	42,100	100%
	Other Images	159,980	100%
MobiStego	MS Stego Images	42,100	100%
	Other Images	159,980	100%
PocketStego	PS Stego Images	42,100	100%
	Other Images	159,980	0.23%

Research funded by the Center for Statistics and Applications in Forensic Evidence (CSAFE) - forensicstats.org

Machine Learning Detection Method

- **Dataset of two case study**
  - 6000+ original images from 3 selected Phone models: half JPEG, half DNG.
  - Cover/stego pairs created from two apps: **PixelKnot** and **Steganography_M**
- **Feature sets**
  - JPEG rich model for frequency domain[1] (for PixelKnot data)
  - Spatial rich model for spatial domain[2] (for Steganography_M data)
- **Classifier**: ensemble FLD[3] (Fisher Linear Discriminant)/Random Forest
- **Average error rate**: (Missed detection rate + False alarm rate)/2

[1] J. Kodovsk`y and J. Fridrich. Steganalysis of jpeg images using rich models. In Media Watermarking, Security, and Forensics 2012.
[2] J. Fridrich and J. Kodovsky. Rich models for steganalysis of digital images. IEEE Transactions on Information Forensics and Security, 2012.
[3] J. Kodovsky, J. Fridrich, and V. Holub. Ensemble classifiers for steganalysis of digital media. IEEE Transactions on Information Forensics and Security, 2012.

Research funded by the Center for Statistics and Applications in Forensic Evidence (CSAFE) - forensicstats.org

Detecting **PixelKnot** stego images*

- No known software package that can test for steganography content in ***mobile phone photographs from stego apps on mobile phones***
- StegoHunt, DC3 cannot detect mobile stego images

Table 5: Error of detection on images generated by PixelKnot

Error Type	Stego Hunt	DC3–StegDetect	Provos–StegDetect
False Alarm	0%	0%	24.6%
Misdetection	100%	100%	75.4%
Avg. Error	50%	50%	50%

- However, **machine learning** trained with proper data – available currently only from StegoAppDB - gives similar performance as in academic setting

Table 6: Classification accuracy of detecting cover-stego pairs by ML algorithms

Apps	Pixel 1	Pixel2	Samsung S8	One Plus 5	Mix of four devices
PixelKnot	97.5%	97.6%	97.6%	98.3%	99.0%
Steganography	98.0%	97.8%	99.4%	97.7%	98.6%
Pocket Stego	96.8%	97.3%	99.5%	98.3%	98.4%
Passlok Privacy	99.0%	97.1%	98.3%	98.3%	98.6%

J. Newman, L. Lin, W. Chen, S. Reinders, Y. Wang, M. Wu, Y. Guan. “StegoAppDB: A steganography apps forensics image database,” IS&T Int’l. Symp. on Electronic Imaging, Media Watermarking, Security, and Forensics 2019, Burlingame, CA, pp. 536-1-536-12, (12), 2019.

Research funded by the Center for Statistics and Applications in Forensic Evidence (CSAFE) - forensicstats.org

Machine Learning Detection Results

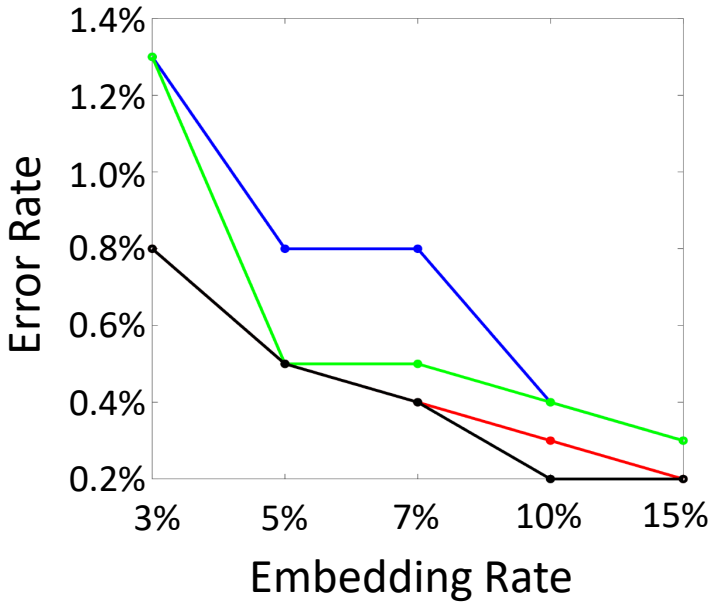
- Detecting stego images created from **Steganography_M**
  - **Spatial domain** embedding with pseudo-random path
  - 850 cover/stego pairs created from 850 center-cropped **PNG** images from **JPEG** images as originals
  - 500 for training, 350 for testing
  - All stegos have 10% embedding rate

Input Image Source	Input Image Size	Stego Image Size	Average Error
Google Pixel	512*512	512*512	0.0%
Samsung Galaxy S7	512*512	512*512	1.0%
OnePlus 5	512*512	512*512	1.4%
Mixed	512*512	512*512	0.8%

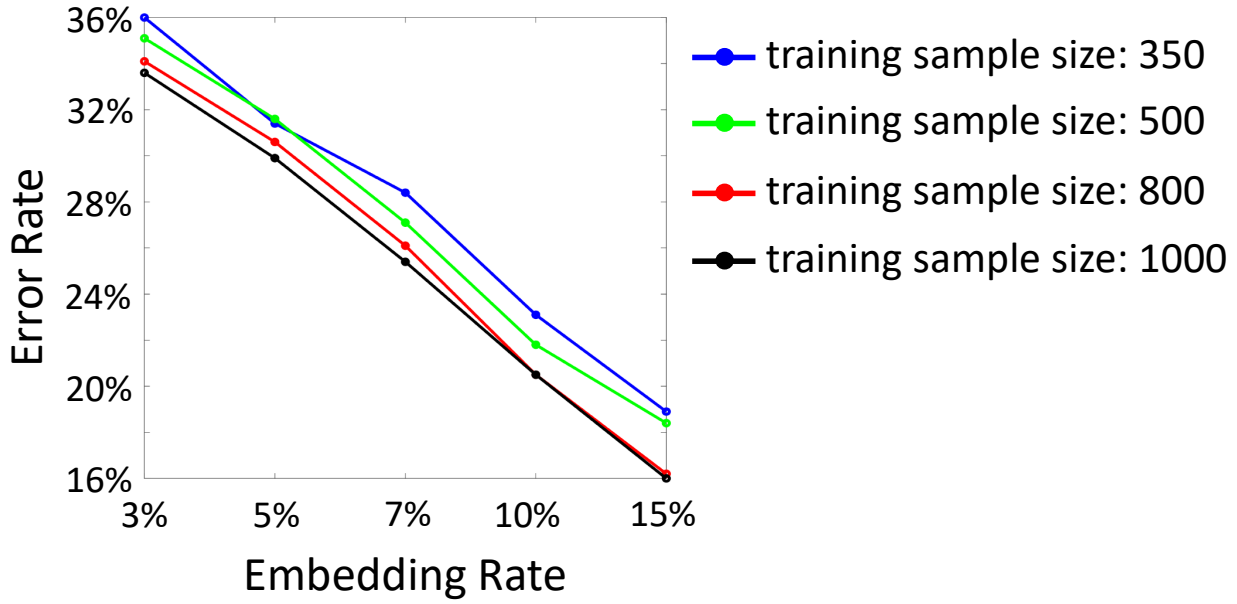
Research funded by the Center for Statistics and Applications in Forensic Evidence (CSAFE) - forensicstats.org

Machine Learning Detection Results

- Detecting stego images created by **Steganography_M,** with different embedding rates, training sample sizes, and original image formats.



Original image format: **JPEG**

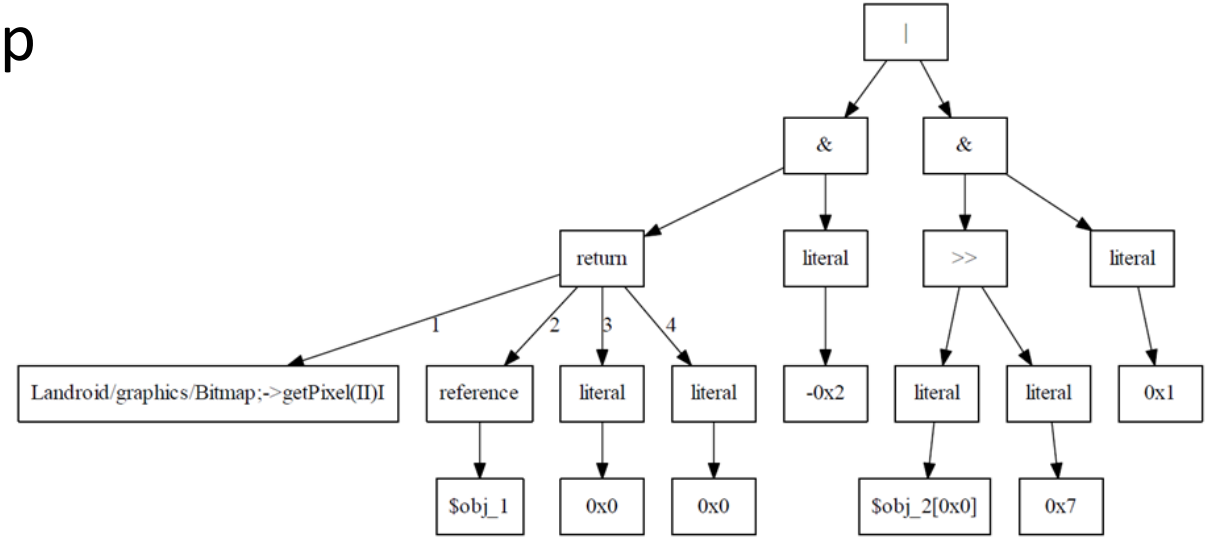


Original image format: **DNG**

Research funded by the Center for Statistics and Applications in Forensic Evidence (CSAFE) - forensicstats.org

Wild Manhunt for Stego apps

- Goal: determine whether any given app (NOT images) contains function/code that performs image steganography.
- Approach:
  - Extract expression trees (ET) by symbolic execution on the app's binary code.
  - Match extracted ETs with a set of pre-defined ETs (domain knowledge) using k-nearest neighbor algorithm
  - The kNN algorithm is based on Tree Edit Distance – the minimum-cost sequence of node edits to transform one tree to another



An example of expression tree, representing how a stego pixel is generated from a cover pixel:

new_pixel = cover.getPixel(0,0) & 0xFFFFFFFE | (p[0]>>7 & 1)

Research funded by the Center for Statistics and Applications in Forensic Evidence (CSAFE) - forensicstats.org

Conclusion

- We generated a mobile stego image dataset by reverse engineering and instrumenting Android stego apps
  - Available at: https://forensicstats.org/resources/datasets-tools/
  - Current analysis process is manual, but future work will focus on automating the procedure to efficiently add new data
- Analysis of stego apps showed embedding signatures existed and can be utilized for high accuracy detection.
- Machine learning can work well without relying on signatures, given access to the devices and cover images.
- A tool for hunting stego apps is still in development, and the progress is encouraging.

Research funded by the Center for Statistics and Applications in Forensic Evidence (CSAFE) - forensicstats.org

Acknowledgements

- The project is funded by CSAFE
- From Iowa State University:
  - Dr. Jennifer Newman and Dr. Yong Guan
  - Li Lin, Stephanie Reinders, Wenhao Chen and Yangxiao Wang
- From University of Maryland:
  - Dr. Min Wu