Data Science in Cyber Security

Projects for the internship
Trista Wang

Two potential topics

01

Steganography

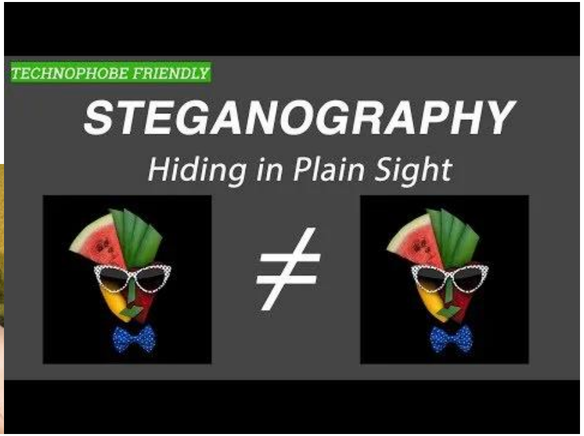
Detection of
hidden data
in images

02

SQL injection

Recognition and
classification
of SQL injection

Detection of steganography: Hidden data in images



Steganography is an evasive technique that aims to conceal a file within another file – in this case, an image – without altering the appearance of the original file to ensure secrecy.

Steganography – the malicious use of digital watermarking

Encryption of malicious content

e.g.: executable PHP codes

Hiding in images

- File metadata
- Spatial domain: hidden in pixels
- Transformed domain:discrete cosine transform (DCT) or discrete wavelet transform (DWT)

Bring in bigger friend

- Concatenation
- A malware to activate the executable files

Steal info from the victim system

Sending commands and information and exfiltrating data.

Steganography – the overview of its mechanism

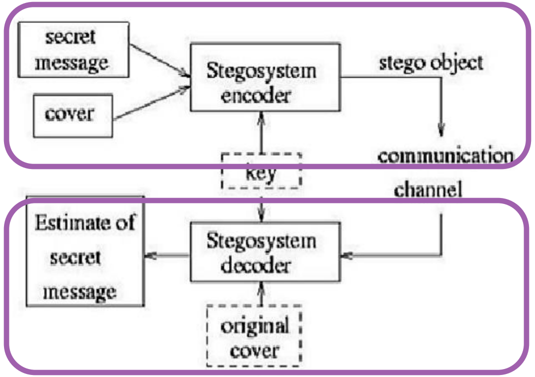


Fig. 1. The basic model of steganography

encoding

decoding

"Steganography is a creative way for hackers to hide what they are doing. It takes advantage of the end-user’s **normal expectations** and **inherent sense of trusting** what we see. "

Key elements:

- Cover image
- Hidden message
- Stego key

Cover image+ hidden message+ stego key = stego image

Data Hidden in Images: Techniques & Detection

What's hidden

- Text message
- Image message
- URL & malicious codes

Steganographic techniques

LSB

JMiPOD
JUNIWARD
UERD
nsf5

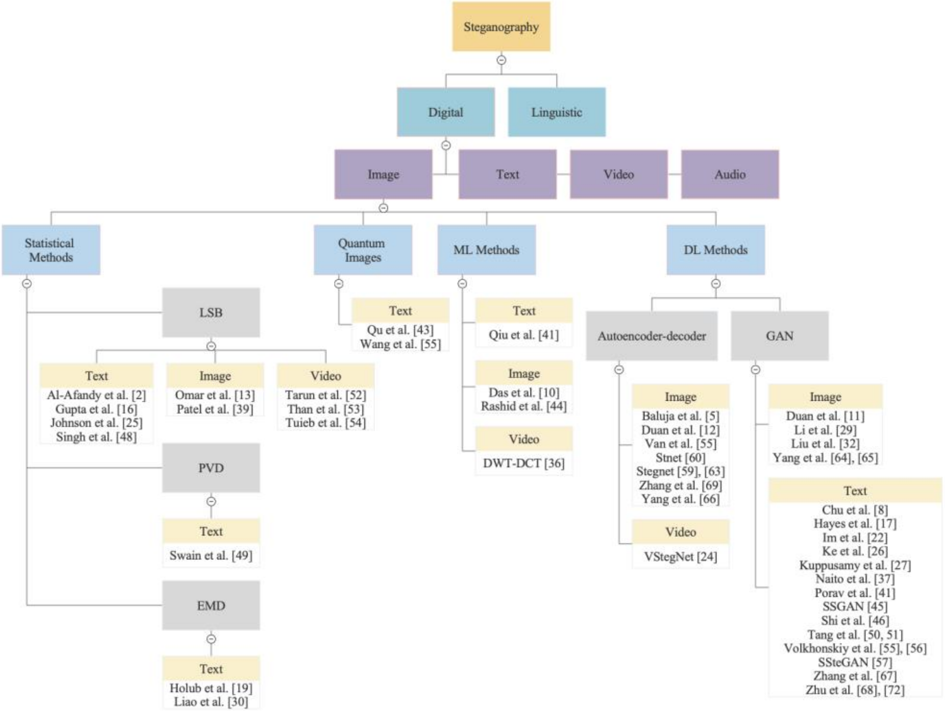
HUGO
WOW

EXIF

Outguess
F5
Steghide
DCT-based

Detection

- Rule-based: String matching...
- Machine Learning
- Deep Learning



Steganography

Digital

Linguistic

Image

Text

Video

Audio

Statistical Methods

LSB

Text

Al-Afandy et al. [2]
Gupta et al. [16]
Johnson et al. [25]
Singh et al. [48]

Image

Omar et al. [13]
Patel et al. [39]

Video

Tarun et al. [52]
Than et al. [53]
Tuieb et al. [54]

PVD

Text

Swain et al. [49]

EMD

Text

Holub et al. [19]
Liao et al. [30]

Quantum Images

Text

Qu et al. [43]
Wang et al. [55]

ML Methods

Text

Qiu et al. [41]

Image

Das et al. [10]
Rashid et al. [44]

Video

DWT-DCT [36]

DL Methods

Autoencoder-decoder

Image

Baluja et al. [5]
Duan et al. [12]
Van et al. [55]
Stnet [60]
Stegnet [59], [63]
Zhang et al. [69]
Yang et al. [66]

Video

VStegNet [24]

GAN

Image

Duan et al. [11]
Li et al. [29]
Liu et al. [32]
Yang et al. [64], [65]

Text

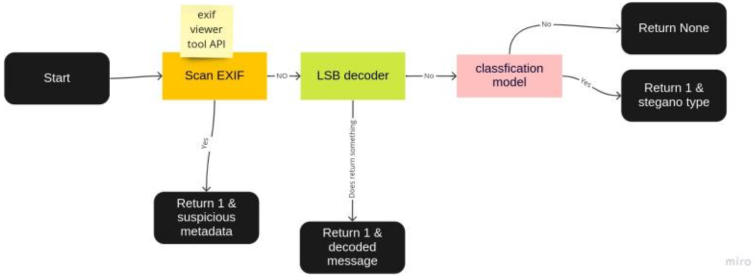
Chu et al. [8]
Hayes et al. [17]
Im et al. [22]
Ke et al. [26]
Kuppusamy et al. [27]
Naito et al. [37]
Porav et al. [41]
SSGAN [45]
Shi et al. [46]
Tang et al. [50, 51]
Volkhonskiy et al. [55], [56]
SSteGAN [57]
Zhang et al. [67]
Zhu et al. [68], [72]

Classification of existent methods based on secret media used

Workflow

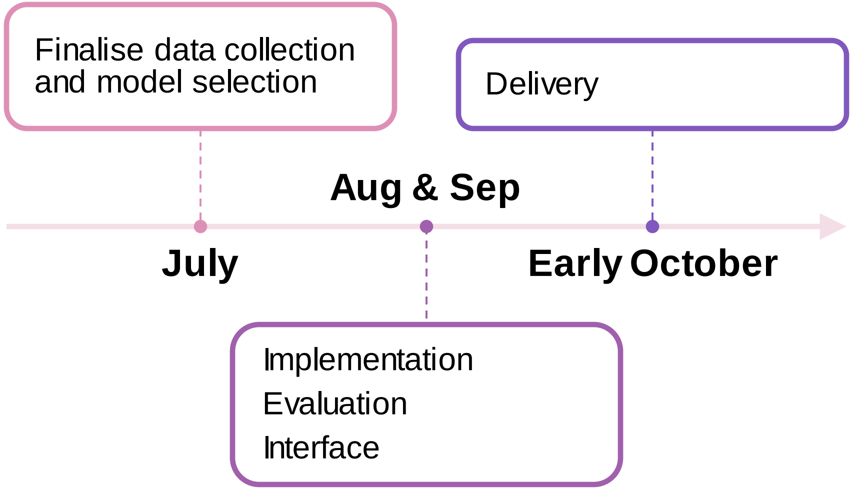
Data Collection	Image data https://data.csafe.iastate.edu/StegoDatabase/ BOSS database	Half done JPEGsteg PNG, BMP and GIF.
Data Processing	• Compression of images • Data augmentation • Kfold splitting	done
Modelling	• Statistical features based • Feature extraction • Architecture selection - EfficientNet - ensemble classifier	done
Evaluation	• Detection rate • False positive rate • Confusion matrix	done

Deliverable & Timeline



Input: JPEG/PNG
Output:
1. 1 or 0: whether there is **steganography**
2. if 1, what's the type of hidden message

Format: An interface to upload the images and return the result



Finalise data collection and model selection

July

Aug & Sep

Implementation
Evaluation
Interface

Delivery

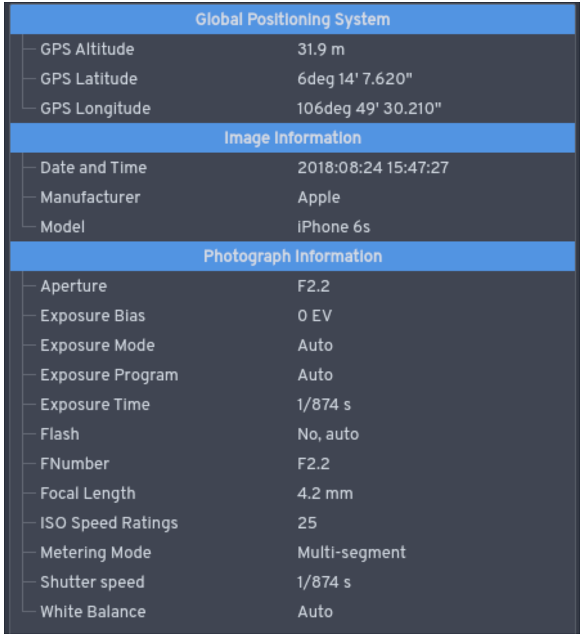
Early October

Q&A

1. With evolving and varied techniques, how to upgrade algorithm or just stacking them?
2. What if we don't know the employed algorithms beforehand (a novel technique)?

For different file formats, can we generalize the model trained on JPEG to other formats?

Strucutally based Stegano – tampering with EXIF



Global Positioning System	
GPS Altitude	31.9 m
GPS Latitude	6deg 14' 7.620"
GPS Longitude	106deg 49' 30.210"
Image Information	
Date and Time	2018:08:24 15:47:27
Manufacturer	Apple
Model	iPhone 6s
Photograph Information	
Aperture	F2.2
Exposure Bias	0 EV
Exposure Mode	Auto
Exposure Program	Auto
Exposure Time	1/874 s
Flash	No, auto
FNumber	F2.2
Focal Length	4.2 mm
ISO Speed Ratings	25
Metering Mode	Multi-segment
Shutter speed	1/874 s
White Balance	Auto

Exchangeable Image FIle Format

Very difficult to detect
60% of the malware in JPEF is introduced through infecting EXIF tags

Tool

Pixeif package

detection (TBD)

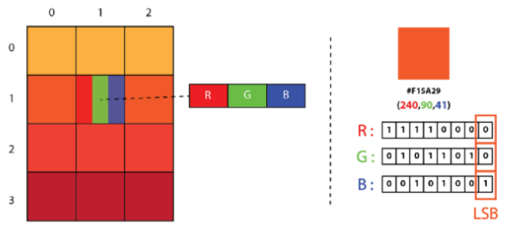
- Exif viewer tool
- Modelling

using the length of the tags as features
Forming TF-IDF

limitations

Cannot identify images without EXIF info

Statistically based Stegno - LSB



Representation of Image as a 2D Array of RGB Pixels

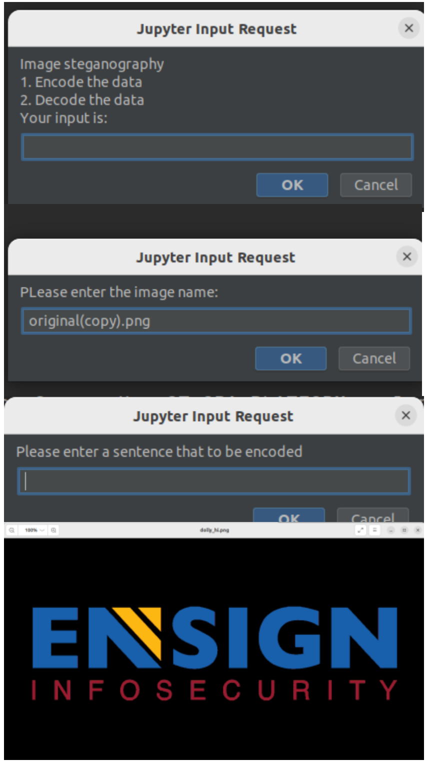

Jupyter Input Request

Image steganography
1. Encode the data
2. Decode the data
Your input is:

Jupyter Input Request

PLease enter the image name:
original(copy).png

Jupyter Input Request

Please enter a sentence that to be encoded

Figure 2. illustrations of LSB implementation

messages are hidden inside an image by replacing each pixel's least significant bit with the bits of the message to be hidden

We can convert the message into decimal values and then into binary, by using the ASCII Table. Then, we iterate over the pixel values one by one, after converting them to binary, we replace each least significant bit with that message bits in a sequence.

StegExpose is a steganalysis tool specialized in detecting LSB (least significant bit) steganography in lossless images such as PNG and BMP. It has a command line interface and is designed to analyse images in bulk while providing reporting capabilities and customization which is comprehensible for non forensic experts.

What is DCT (Discrete Cosine Transform)

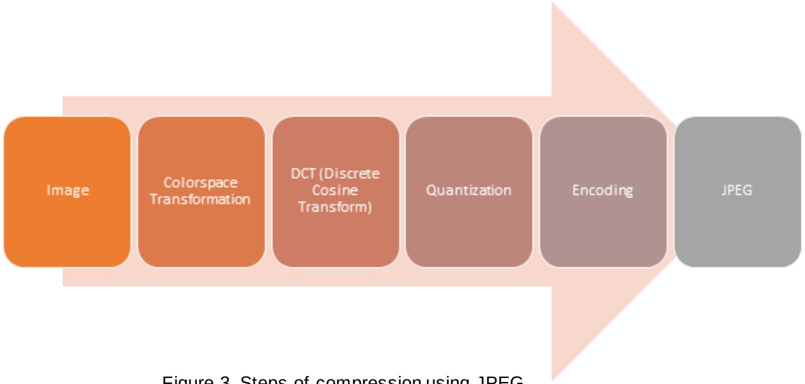
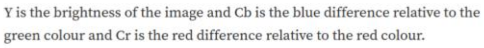
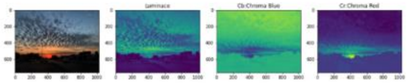


Figure 3. Steps of compression using JPEG

Colorspace transformation:
RGB to YCbCr



Y is the brightness of the image and Cb is the blue difference relative to the green colour and Cr is the red difference relative to the red colour.



DCT+quantization:

The way that the discrete cosine transform works, is we take some data, in this case, our image data, and we try to represent it as the sum of lots of cosine waves. It transfers an image from the spatial domain to a frequency domain.
essentially removes the high frequency information in image

Video

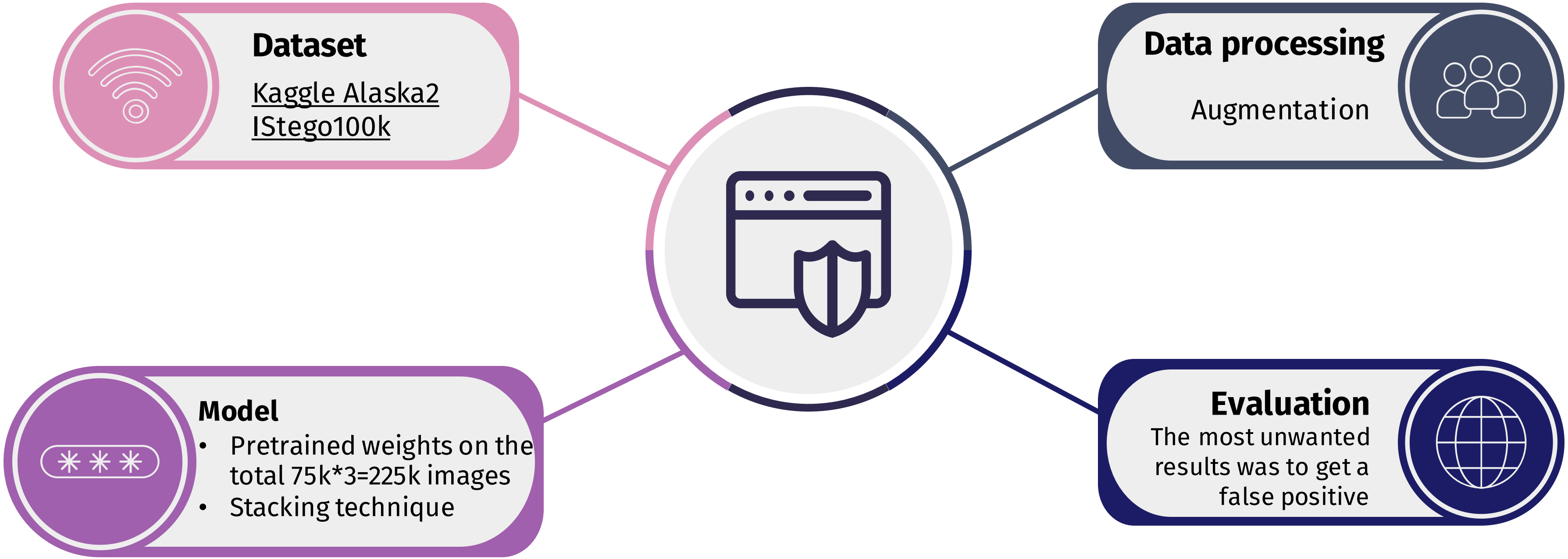
Huffman encoding

An algorithm to improve space efficiency of saving information in binary

Video

DCT based Stegno - Modelling

Detection of JMiPOD, JUNIWARD and UERD, NSF5 embedding methods



Dataset

Kaggle Alaska2
IStego100k

Data processing

Augmentation

Model

- Pretrained weights on the total 75k*3=225k images
- Stacking technique

Evaluation

The most unwanted results was to get a false positive

Approximate training time of the whole ensemble is ~4 weeks on 3xTitan RTX

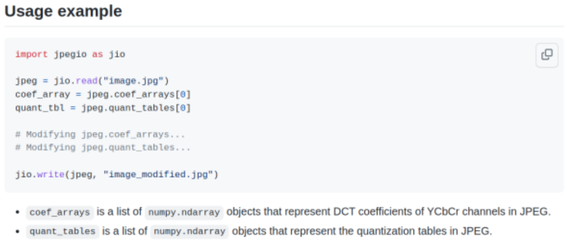
DCT based Stegno - Modelling

Detection of JMiPOD, JUNIWARD and UERD, NSF5 embedding methods

First_stage models

Steganalysis features

Jpegio:A python package for accessing the internal variables of JPEG file format such as DCT coefficients and quantization tables



Usage example

```
import jpegio as jio

jpeg = jio.read("image.jpg")
coef_array = jpeg.coef_arrays[0]
quant_tbl = jpeg.quant_tables[0]

# Modifying jpeg.coef_arrays...
# Modifying jpeg.quant_tables...

jio.write(jpeg, "image_modified.jpg")
```

- `coef_arrays` is a list of `numpy.ndarray` objects that represent DCT coefficients of YCbCr channels in JPEG.
- `quant_tables` is a list of `numpy.ndarray` objects that represent the quantization tables in JPEG.

EfficientNet models

Different choices of EfficientNet B2, B4, B5

MixNet models

MixNet_S, MixNet_XL

second_stage stacker

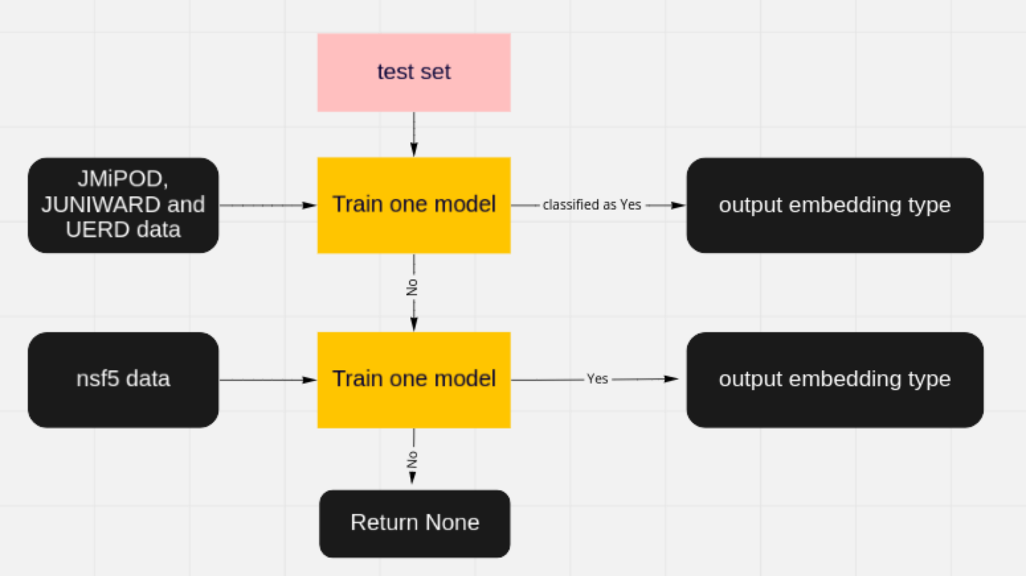
Voting results from above models are input features

Catboost Classifier

Statistically based Stegno - modelling

Detection of JMiPOD, JUNIWARD and UERD, NSF5 embedding methods

Due to the differences in image source/size

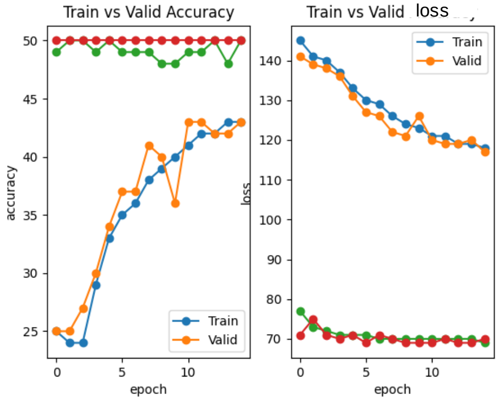


Due to the differences in image source/size

However, can only find pretrained weights on first model. Thus, the performance on nsf5 classification is not satisfying yet.

Model results

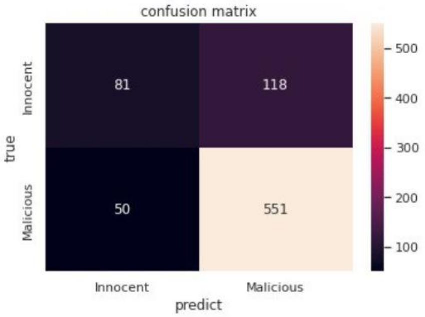
V0



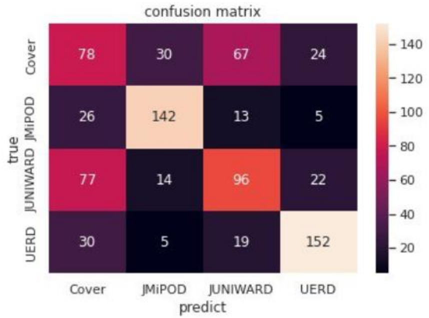
Data: 8k images for training, 2k for validation

Model: EfficientNet B2 only

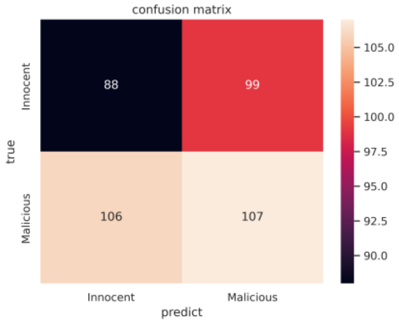
V1



	Innocent	Malicious
Innocent	81	118
Malicious	50	551



	Cover	JMiPOD	JUNIWARD	UERD
Cover	78	30	67	24
JMiPOD	26	142	13	5
JUNIWARD	77	14	96	22
UERD	30	5	19	152



	Innocent	Malicious
Innocent	88	99
Malicious	106	107

For UERD, JUNIWARD, JMiPOD

Model: catboost classifier as second-stage model and pretrained weights on a much larger image set

For nsf5

Data: 12k for training

Current issues & Future steps

Supported format

Only cover JPG & PNG
LSB decoder: PNG only
DCT-based stegano classifier: JPEG only

--- Try to expand to other formats

Model performance

- Train models on nsf5 with larger set
- Innocent images tend to be predicted as malicious

--- experiment with counter-unbalance measures on Catboost/ look into how Cover and JMiPOD can be differentiated

--- Check confusion matrix of different models, perhaps for those with raw images (RGB values) as input, EfficientNet and MixNet, they are adding confusion, leading to false positive. If yes, try to change the input instead of raw images.

Think about how the results should be interpreted in real cases?
- if detected as malicious, what can be the following steps?

Thoughts & Takeaways

For image processing:

- Think twice about resizing and standardization
- Think twice before using hard image augmentations
- Any better alternatives than inputting raw images themselves? -- Based on the question itself, rely on feature engineering
- Be cautious about file format requirements before starting the project

Overall:

- The choice of evaluation methods can also influence the model performance
- In research phase, be prepared for more than one solution or direction
- Think about generalization, how the project can be used for more realistic scenarios. (introduce noise/different image sources for robustness)

Thank you