Active forensics

framework

Source identification using ***Boost algorithm***

Authors

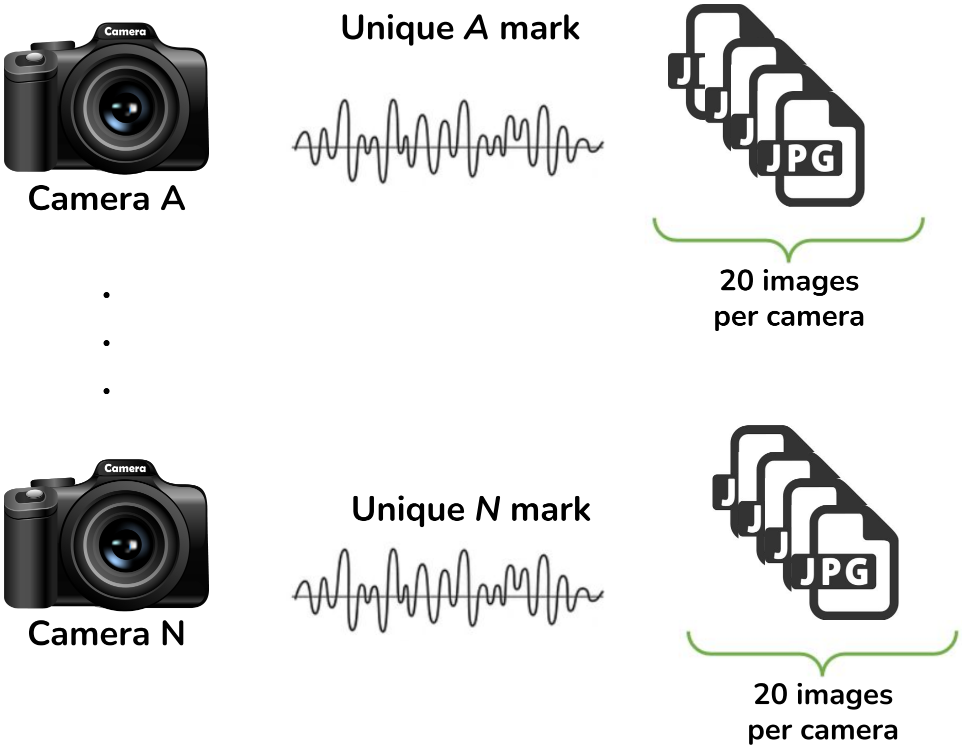
Gabriele Motta
Michele Panariello
Nicola Arpino

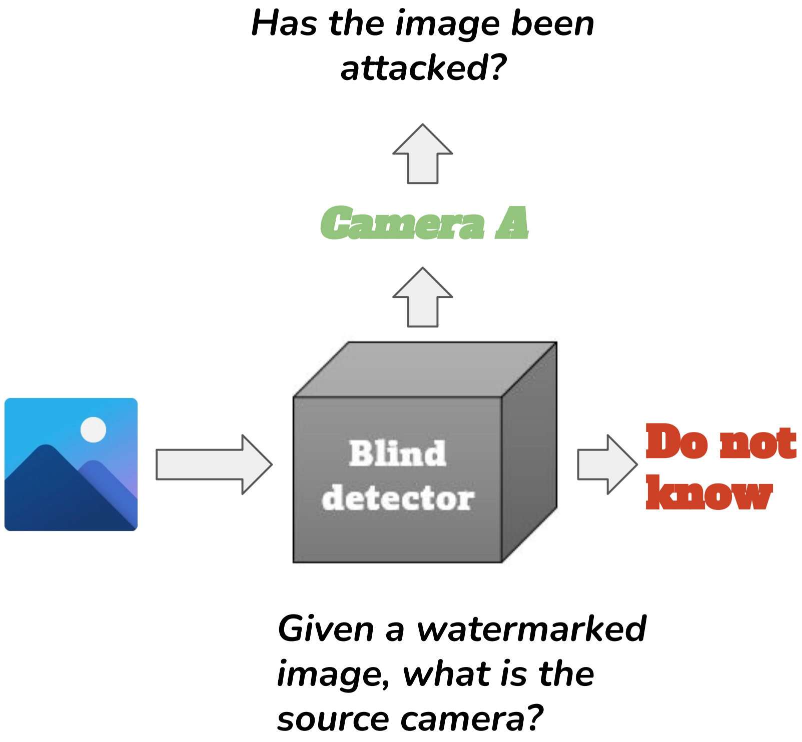
Index

BOOSTALGORITHM

Introduction: What problem we have to address?

Bijective correspondence between devices and marks





Introduction: What problem we have to address?

Starting point:

- 5 camera
- 20 images/camera

Constraints

We have identified some major points:

1. ***Marks must be dissimilar from each other.***
2. ***One mark per device.***
3. ***Detector is blind.***

So, how we do choose the 5 marks?

Answer: Distribution of matching bits ***fairly as possible***.

1	1	1	1	1	0	1	0	0

Method: **Fair Mark Generation**

Fair Mark Generation

- Marks are binary strings.
- Marks different as possible from each other.
- TPR maximized.

Let us see the case with 5 marks, that is indeed our case.

1	1	1	1	0	0	0	0	0	0	Mark 1
1	0	0	0	1	1	1	0	0	0	Mark 2
0	1	0	0	1	0	0	1	1	0	Mark 3
0	0	1	0	0	1	0	1	0	1	Mark 4
0	0	0	1	0	0	1	0	1	1	Mark 5

All the accuracies between every pair of marks is equal to 40%.

$$\frac{\binom{m-2}{\frac{m}{2}-2} + \binom{m-2}{\frac{m}{2}}}{\binom{m}{\frac{m}{2}}}$$

Fair mark generation

Pro

- TPR is truly maximized.
- Extensible in marks’ size.
- Marks have 40% of 1’s.
- Any fair mark set has maximum accuracy less than 0.5.

Cons

- Minimum amount of bits needed to have a set of fair marks is $\binom{m}{\lfloor \frac{m}{2} \rfloor}$
- The length of the marks must be a multiple of previous quantity.
- Not extensible in marks’ number.

Random Mark Generation

Pro

- More scalability.
- Extensibility.
- No bounds on the mark length.

Cons

- True positive rate not maximized.
- Far from optimality.
- Maximum accuracy larger than 0.5.

Embedding

1. *Calculate the optimal size of the chunk via* ***adaptive approach***

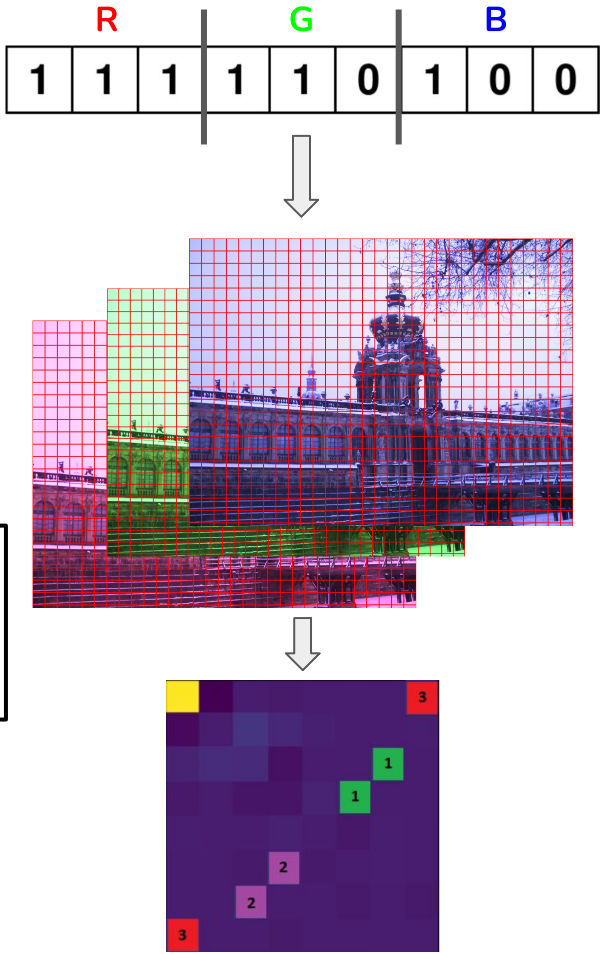
2. *Calculate the portion of the mark that goes into each image layer*

3. *Retrieve all the chunks from the each layer*

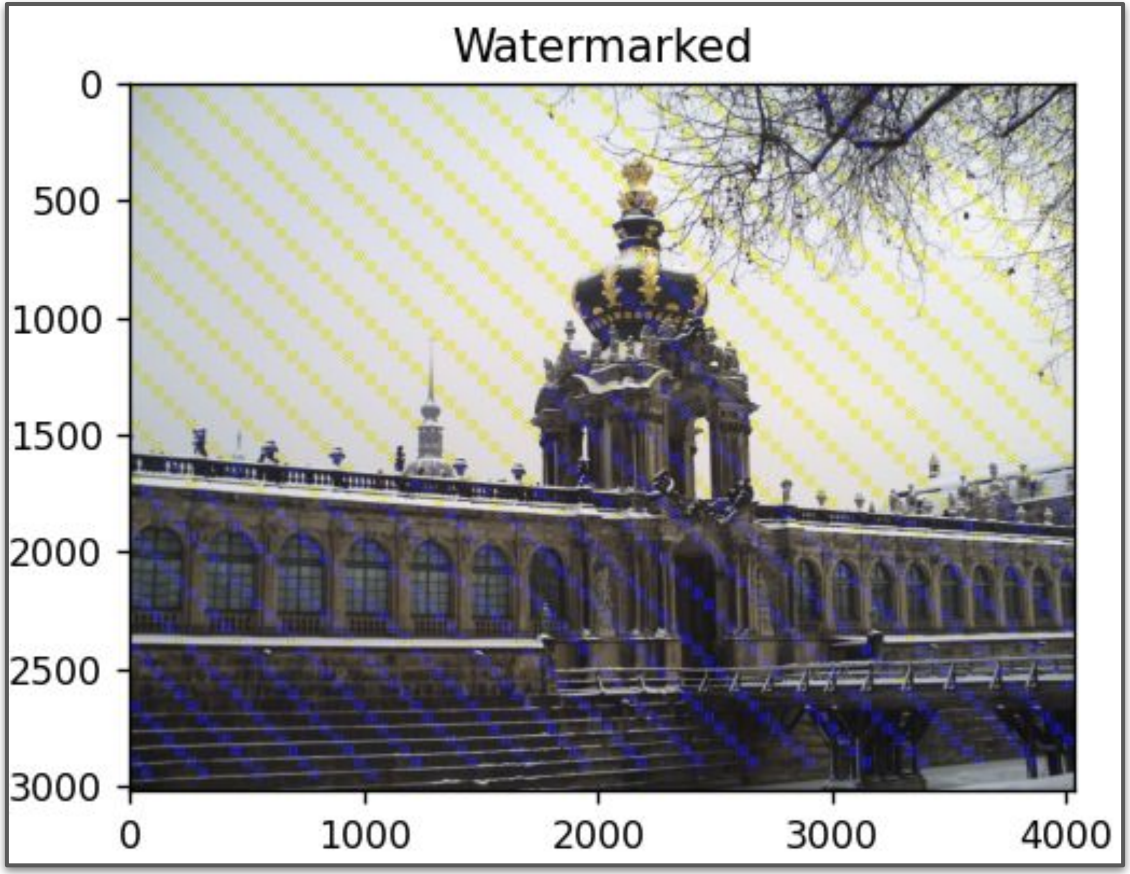
4. *Cycle over chunks of each layer and follow the* ***partitioning*** *and* ***embedding rule***

Partitioning rule:
We further divide the mark in 3 portions. When we embed the first subarray we use the pair of coefficients of indices *[(2, 6), (3, 5)]*, for the second one we use *[(5, 3), (6, 2)]* and finally we use the pair *[(0, 7), (7, 0)]*

Embedding rule:
Each DCT chunk uses the ***partitioning rule***
If we want to embed *0* boost the first spot, else boost the second one. To such a spot is also added the absolute difference between the two.



Embedding: example



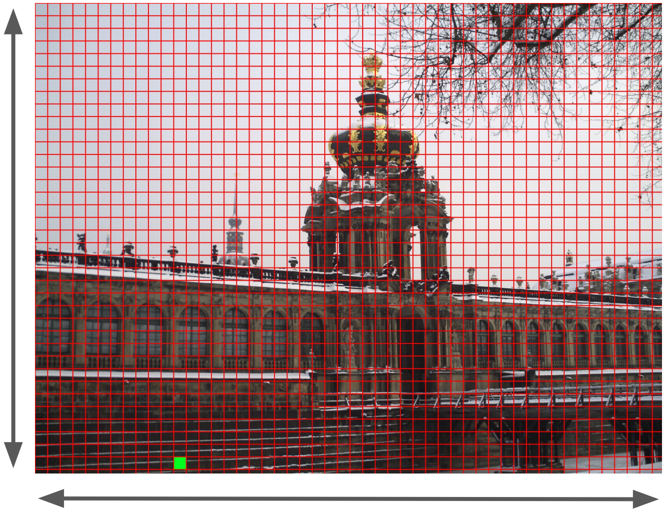
Detection

1. *Calculate the optimal size of the chunk via **adaptive approach** considering the image width and height*

2. *Calculate the portion of the mark embedded into each image layer*

3. *Retrieve all the chunks from each layer*

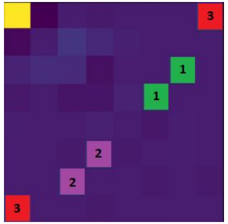
4. *Cycle over chunks of each layer and follow the **extraction rule***



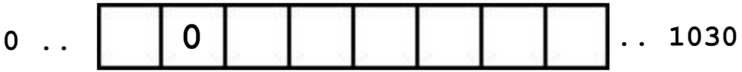
Extraction rule

For a given pair of spots, selected according to partitioning rule, check where is the max value, ***if the max value is in the first spot extract a zero, else a one***

Apply DCT to chunk



`Extracted_mark[i]`



Identification process

How it works?

1. Compare the extracted mark pairwise with each other available original mark using ***accuracy*** as metric.

2. The comparisons with highest and lowest accuracies are saved

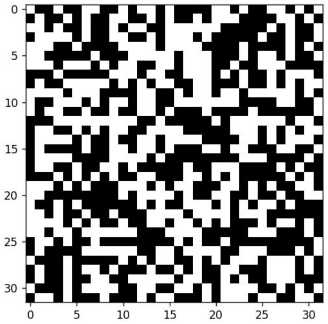
3. Comparison with lower and upper thresholds

4. Select the most suitable accuracy

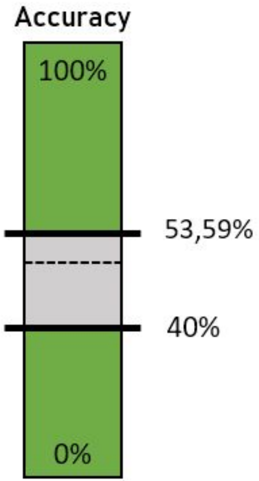
Tampering detection

The identifier is also able to state, in case the mark is present, if the original image has been attacked. Accuracy <= 97%

Extracted Device-0 mark retrieved from the detection



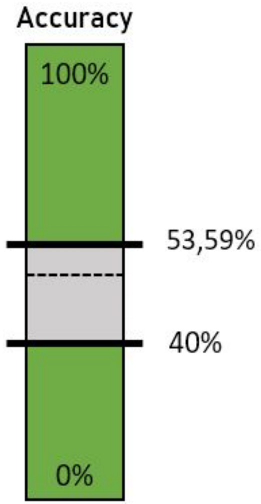
Original marks of devices 0 -> 4



Pairwise accuracies

[57.28, 47.96, 49.71, 52.23, 47.18, 47.48]

Accuracy between Device-0 original mark and the extracted one after an attack is the highest one!



How the bounds have been calculated?

- **Upper-bound**: 53,59%

  The value has been obtained after multiple ROC iterations using many attacks with different level of energy

- **Lower-bound**: 40%

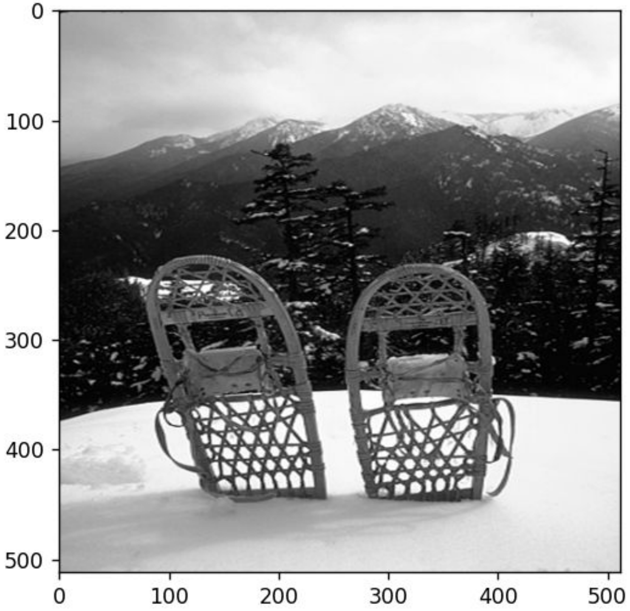
  This value comes from the Fair marks generation. A mark is “similar” to each other one with an accuracy of 40%.

Before presenting our results

Structural similarity index (SSIM) is the metric used. Based on the ***Human visual perception system***.

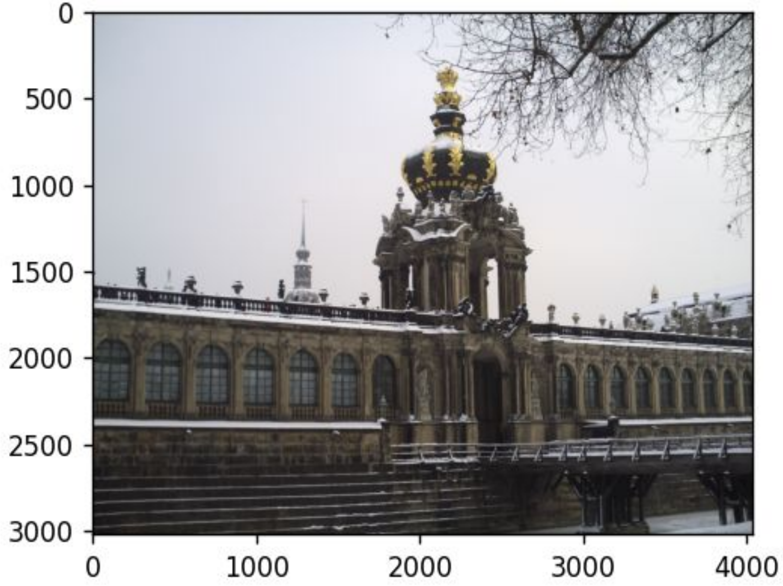
Typical values range is ***[0,1]***. *Value 1 means very similar.*

Watermarked image
Gray scale



***WPSNR**: 63.20*
***SSIM**: 0.9956*

Watermarked image
RGB



***SSIM**: 0.9953*

Good embedding quality >= 0.995 --- Destroyed image < 0.9

SSIM Explanation

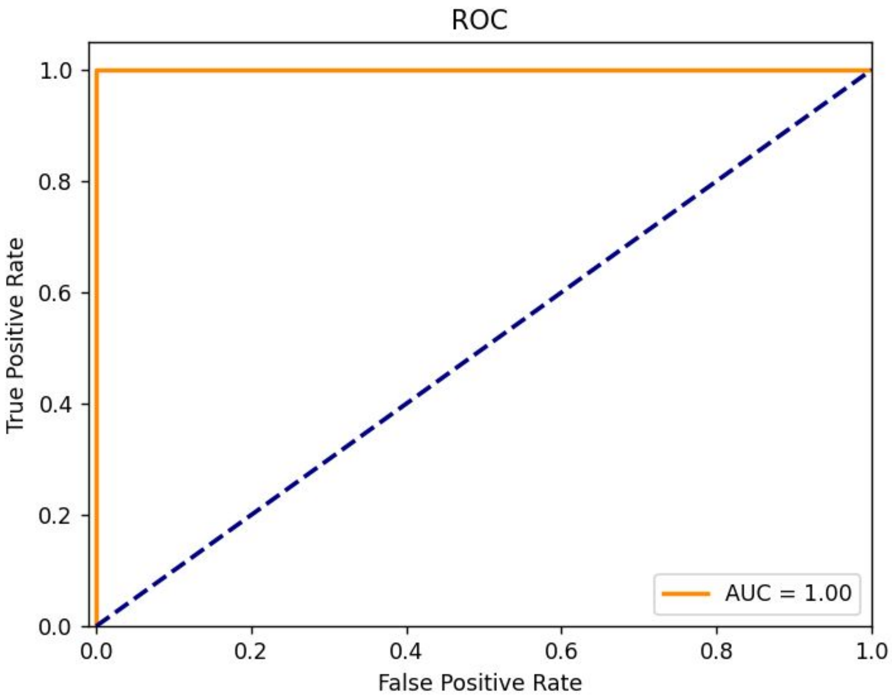
The SSIM is calculated based on three parameters: l(x, y) (luminance comparison), c(x, y) (contrast comparison) and s(x, y) (structure comparison).

$$\mathrm{SSIM}(\mathbf{x},\mathbf{y}) = [l(\mathbf{x},\mathbf{y})]^{\alpha} \cdot [c(\mathbf{x},\mathbf{y})]^{\beta} \cdot [s(\mathbf{x},\mathbf{y})]^{\gamma}$$

- **Luminance:** Luminance is measured by *averaging* over all the pixel values.
- **Contrast**: It is measured by taking the *standard deviation (square root of variance)* of all the pixel values.
- **Structure**: Based on structure properties of objects in the scene.

For those details we refer to the resource below
https://www.cns.nyu.edu/pub/eero/wang03-reprint.pdf

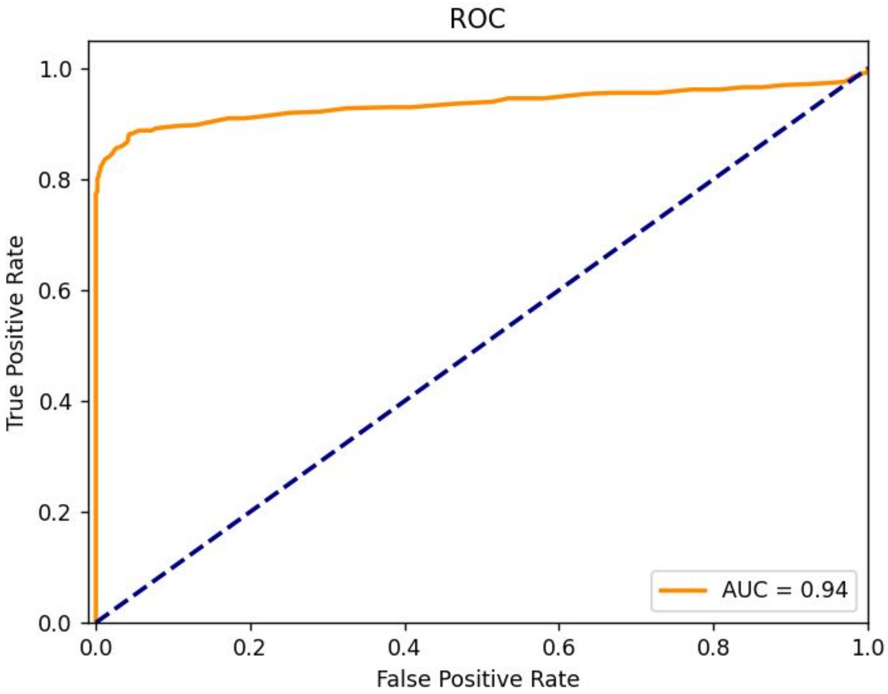
Results: ROC without attacks



ROC has been computed using all the images per device (100 total images). Those images have ***not been attacked.***

The classifier is as good as the "ideal" one.

Results: ROC with attacks



RESULT: *FPR = 0.05* ⇒ *TPR = 0.87*

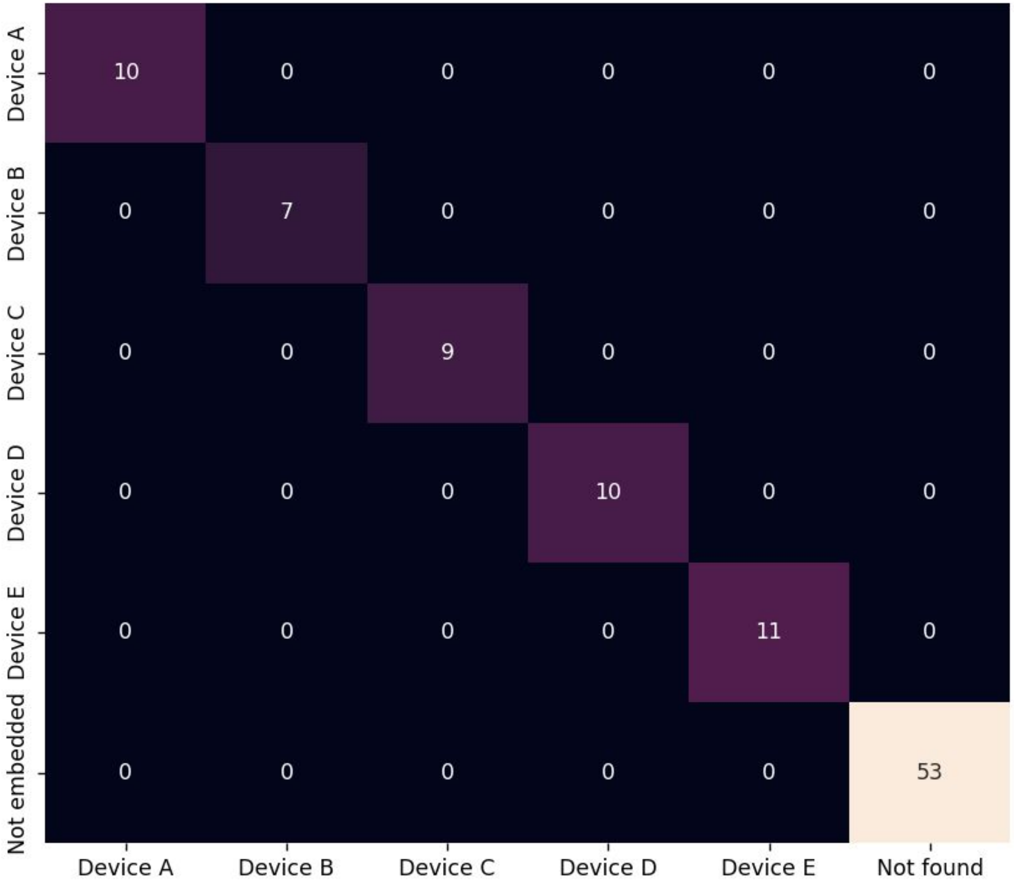
ROC has been computed using all the images per device (100 total images). Those images have been attacked 10 times with all the available attacks configured as follows:

- **Gaussian Noise:** (AWGN) sigma = 300
- **Blur**: sigma = 6
- **Sharpening** = sigma = 10; alpha = 10
- **Resizing**: 0.10
- **Jpeg compression**: QF = 13

Results: confusion table without attacks

Confusion matrix is based on statistical data per-device like the following:

Device: A
Embedded: 10/20
Attacked: 0/20
True positive rate: 10/10 => 100.0
False positive rate: 0/10 => 0.0
Accuracy: 20/20 => 100.0



Results: confusion table with attacks

Attacks settings
Gaussian noise: sigma = 150
Blur: sigma = 4
Sharpening: sigma = 8, alpha = 8
Median: kernel = 21
Resizing: scale = 0.15
Jpeg compression: QF = 15

	Device A	Device B	Device C	Device D	Device E	Not found
Device A	5	0	0	0	0	0
Device B	0	10	0	0	0	1
Device C	0	0	3	0	0	0
Device D	0	0	0	8	0	2
Device E	0	0	0	0	12	0
Not embedded	0	0	0	0	0	59

What did not work

- Different embedding and detection based on uniformity or non uniformity of the coefficients.
- Sorted chunks by **average** or **variance**.
- Fixed chunk size

Conclusions

Strongnesses

- Boost strength and embedding ratio are tunable.
- High robustness and quality.
- Low mismatch ratio.
- Double threshold.
- Fair mark generation.

Weaknesses

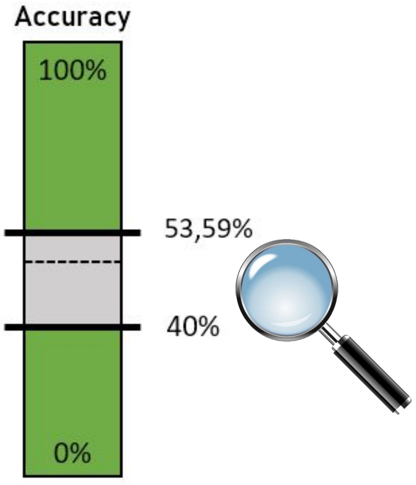
- The amount of bit space needed grows fast, as the number of device grows.
- Lightweight attacks tampering not detected.

Future works

- Tests with combined standard attacks
- Tests with geometric attacks
- Tests with a larger dataset
- Embedding using bigger chunks
- Gray area analysis

Future works: identification inside the **grey area**

Can we do better?



Idea:

Exploit of the information given by all accuracies pairs using mathematical tools like **average** and **standard deviation.**

Example:

Device embedded: 0

If the maximum accuracy is greater than the accuracies average excluded the greatest ***+ 5****, select the greatest accuracy.*

Greatest accuracy: 52.58
Accuracies avg: 47,352 (excluded the greatest)

52.58 > 47,352 + 5 (yes)

[52.58, 47.00, 46.78, 47.50, 48, 47.48]

Summary

Problem: blind device identification and tampering detection

Solution proposed:

- Fair mark generation
- embedding: increment the value of one of two dct chunk spots
- detection: check the greater dct chunk spot value between two spots
- identify: check the most suitable accuracy using the double threshold method

Results: ROC -> **0.87 TPR** with **0.05 FPR**

Future improvement: grey area analysis