UNIVERSITY OF TRENTO - Italy
Department of Information Engineering and Computer Science

An Image-statistics-based approach to detecting Recaptured Images

Signal, Image and Video

Corso Laura 230485
Miotto Sara 232086
Zaghen Olga 224436

Our reference paper

A Simple and Effective Image-Statistics-Based Approach to Detecting Recaptured Images from LCD Screens

Kai Wang - Univ. Grenoble Alpes, CNRS, Grenoble INP, GIPSA-lab, 38000 Grenoble, France

Goal of the project

The project aims to distinguish between single captured and recaptured images using a statistic-based discriminative approach through an SVM classifier.

A single captured image is a real-world scene captured by a digital camera.

A recaptured image is a single captured image displayed on a LCD screen that is then recaptured by a digital camera and eventually tampered.

Single Captured images

Recaptured images

Two recaptured tampered images on which, from up to bottom, color alteration and object removal have been respectively used for creating the fake image.

Images from the ROSE database (Cao and Kot, 2010; Cao, 2010).

Pipeline of the adopted method

1

Pre-processing

Conversion to B&W and resizing of the image (width of 2048 pixels).

2

Computation of the residue images

3

Pixel-wise correlation coefficients

4

Feature Vector

Removing of redundancy and concatenation.

5

Classification

Pipeline of the adopted method

1

Pre-processing

Conversion to B&W and resizing of the image (width of 2048 pixels).

2

Computation of the residue images

3

Pixel-wise correlation coefficients

4

Feature Vector

Removing of redundancy and concatenation.

5

Classification

Pipeline of the adopted method

1

Pre-processing

Conversion to B&W and resizing of the image (width of 2048 pixels).

2

Computation of the residue images

3

Pixel-wise correlation coefficients

4

Feature Vector

Removing of redundancy and concatenation.

5

Classification

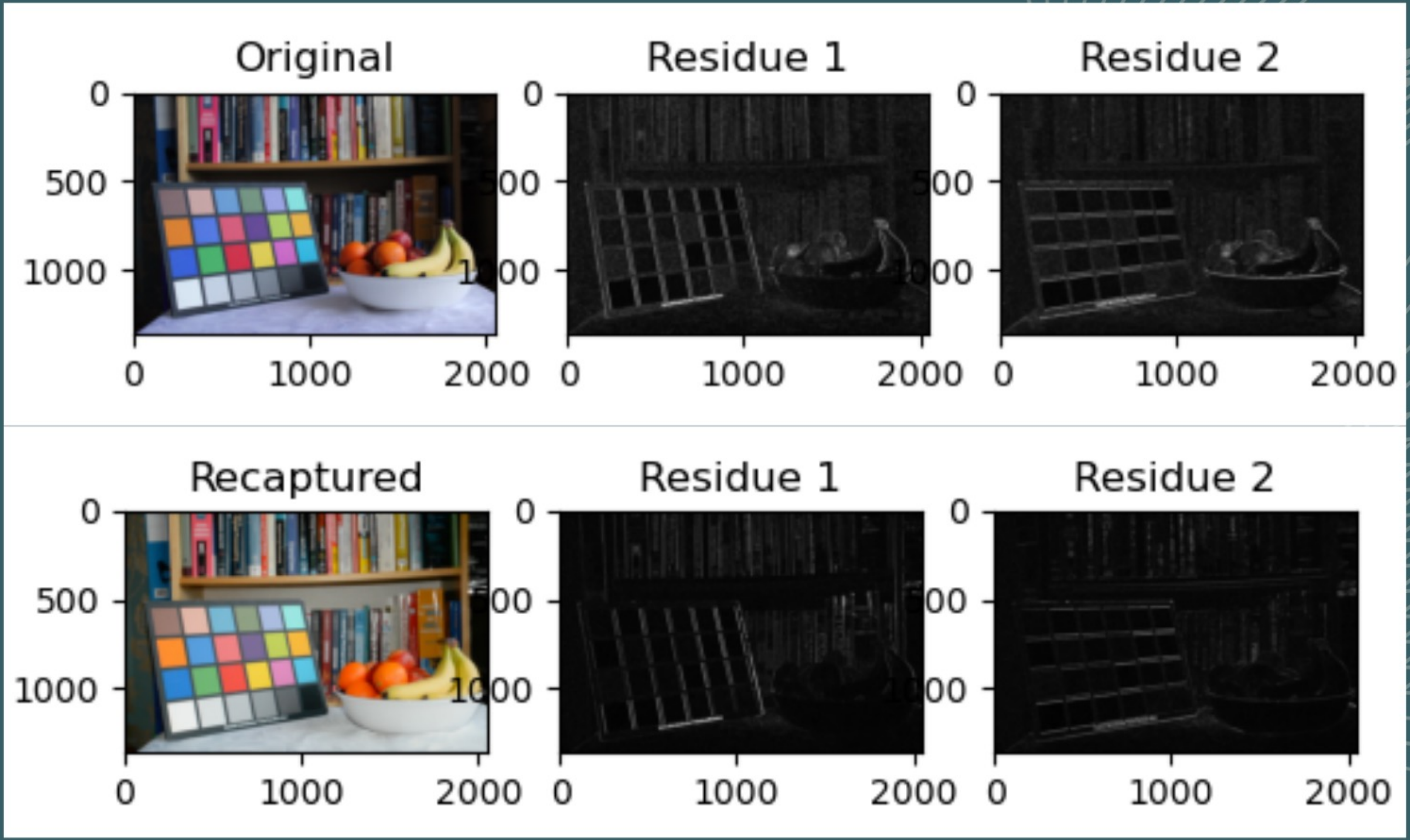
$$\begin{bmatrix} 0 & 0 & 0 \\ \frac{1}{2} & 0 & \frac{1}{2} \\ 0 & 0 & 0 \end{bmatrix}$$

LP *f1*

$$\begin{bmatrix} 0 & \frac{1}{2} & 0 \\ 0 & 0 & 0 \\ 0 & \frac{1}{2} & 0 \end{bmatrix}$$

LP *f2*

*R = trim(X - X*fi)*



The residue images have very different characteristics due to alterations introduced by image recapture. For a better visualization, we have taken pixel-wise absolute values of the residue images and performed thresholding at 15 (the range of pixel values in unfiltered images is from 0 to 255).

Pipeline of the adopted method

1

Pre-processing

Conversion to B&W and resizing of the image (width of 2048 pixels).

2

Computation of the residue images

3

Pixel-wise correlation coefficients

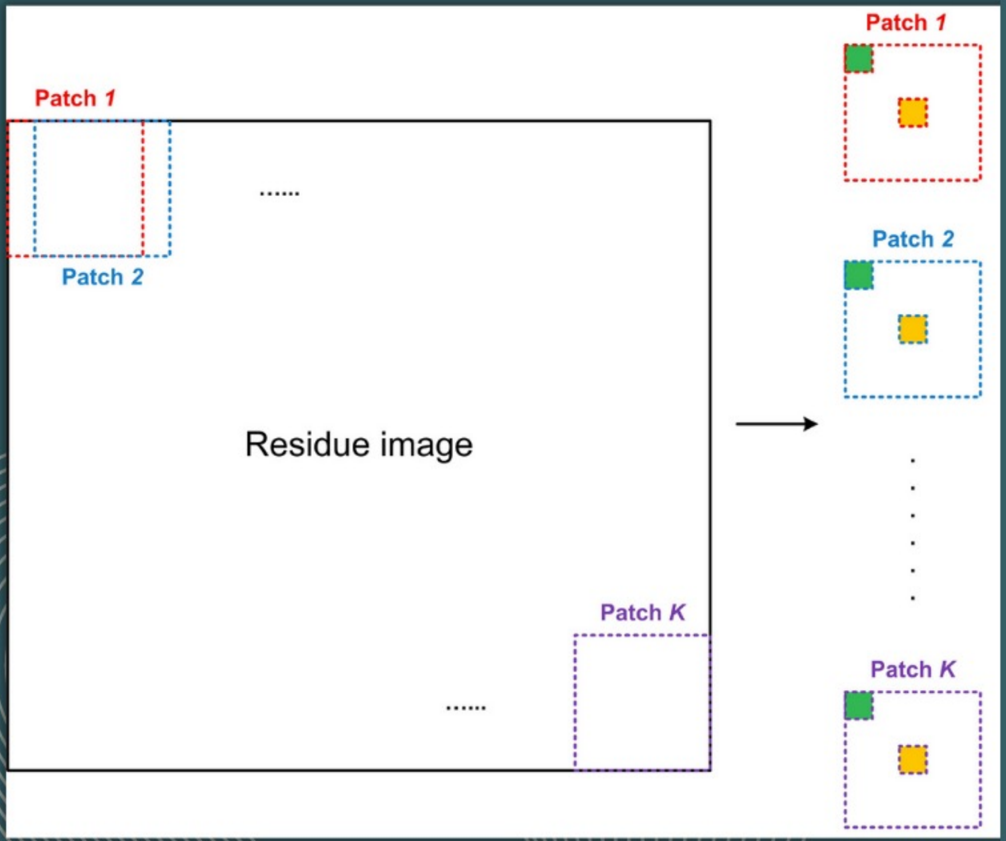
4

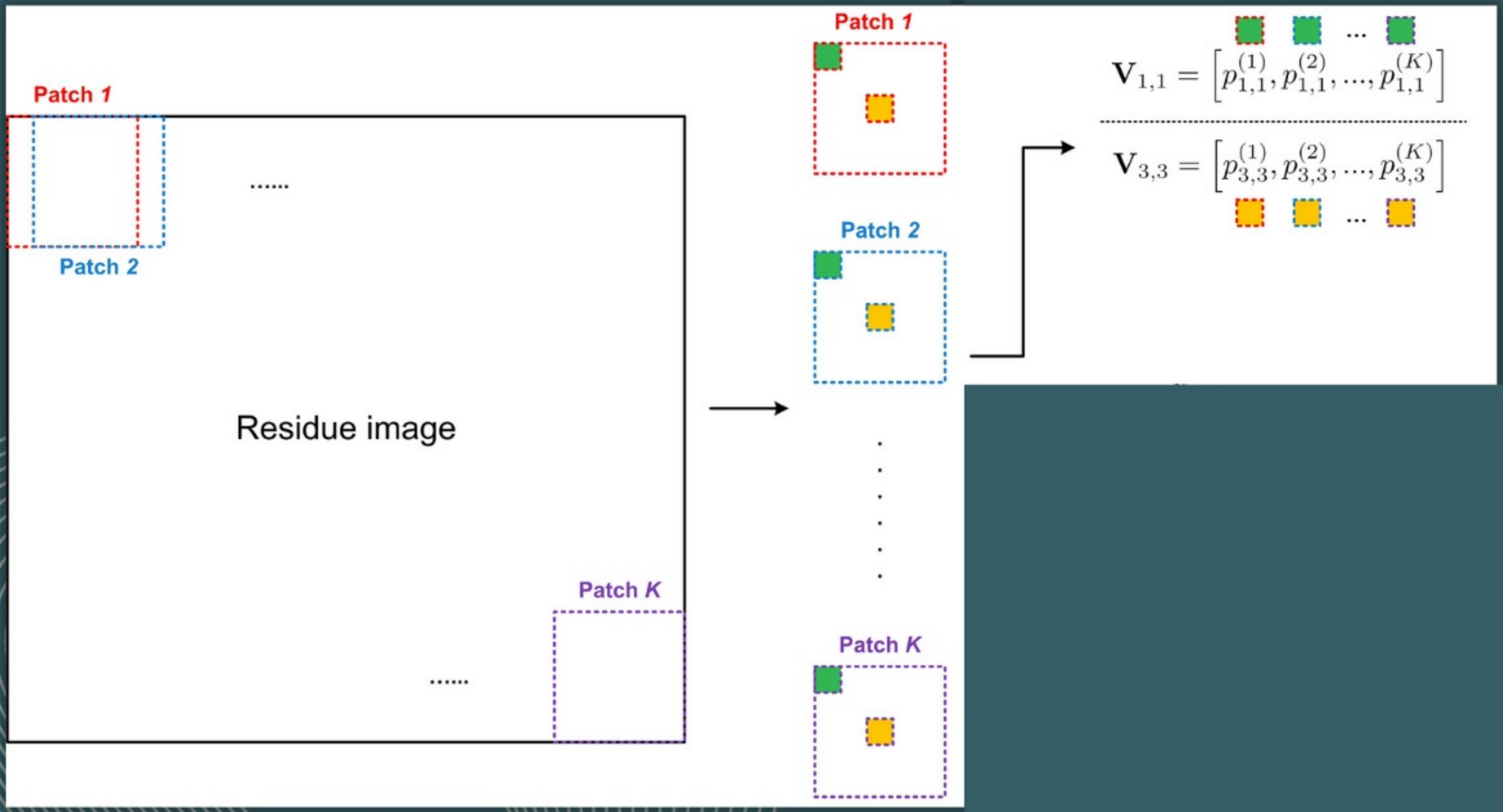
Feature Vector

Removing of redundancy and concatenation.

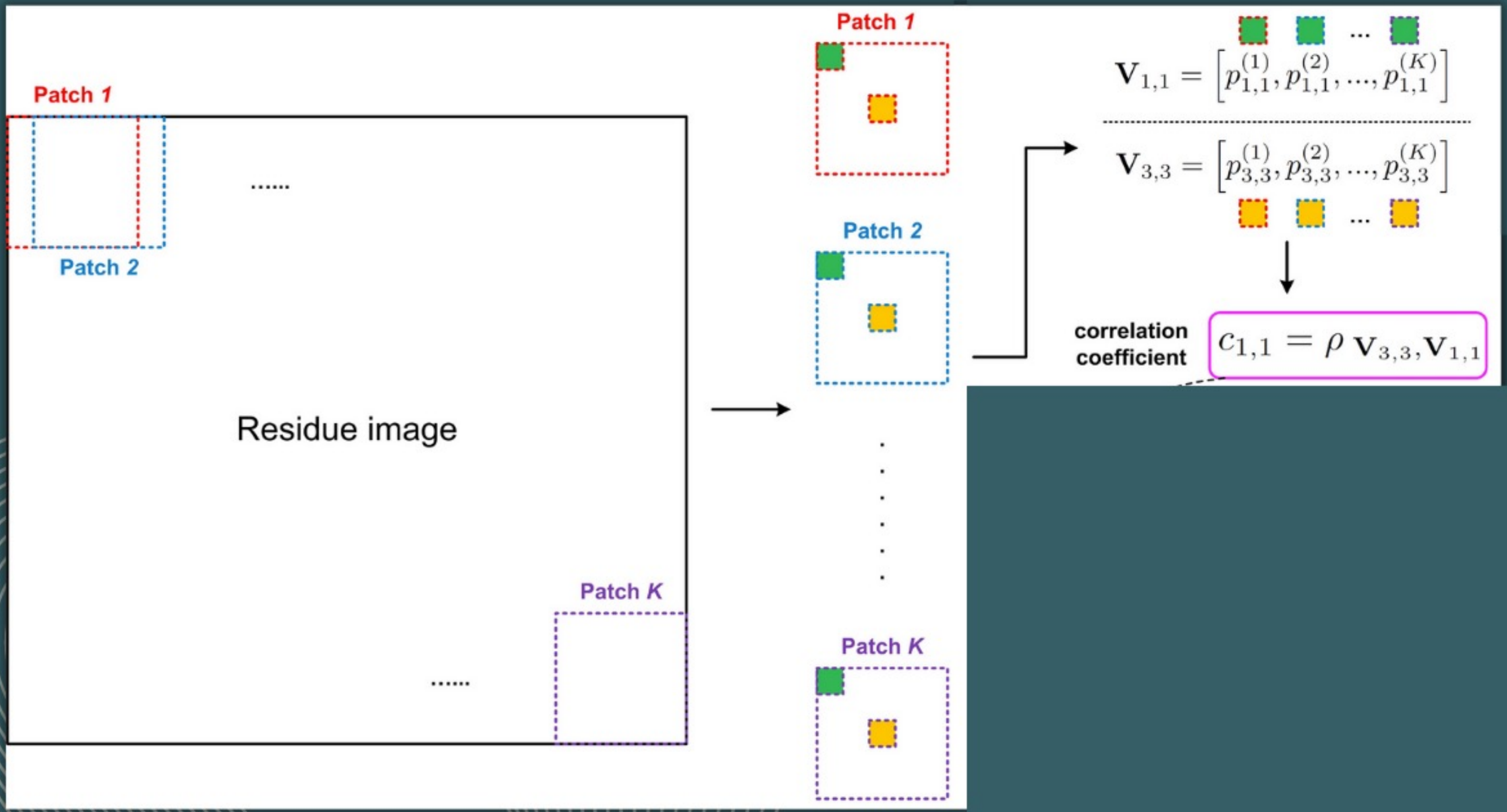
5

Classification

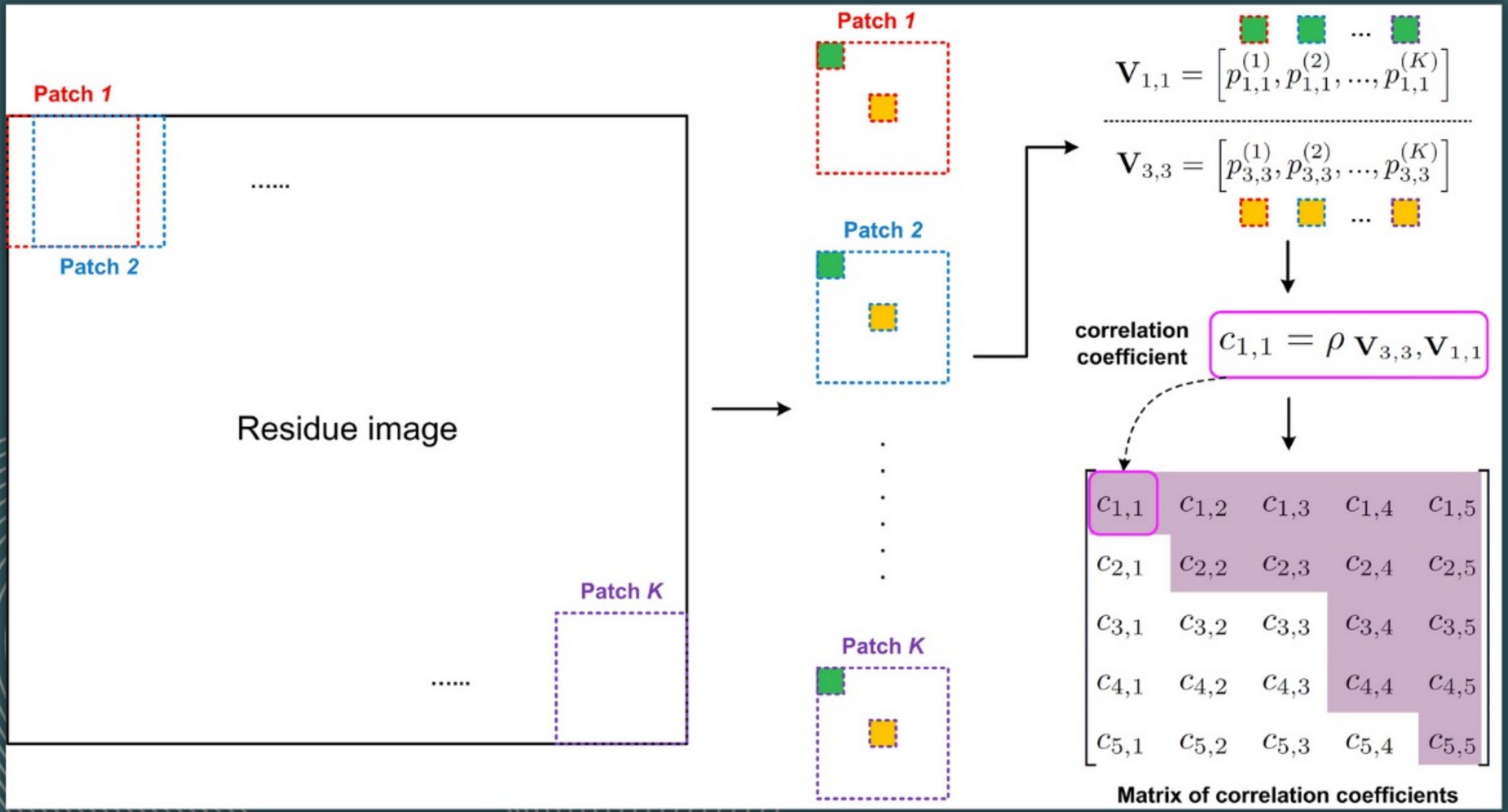

Patch 1
Patch 2
......
Residue image
Patch K
......
Patch 1
Patch 2
Patch K



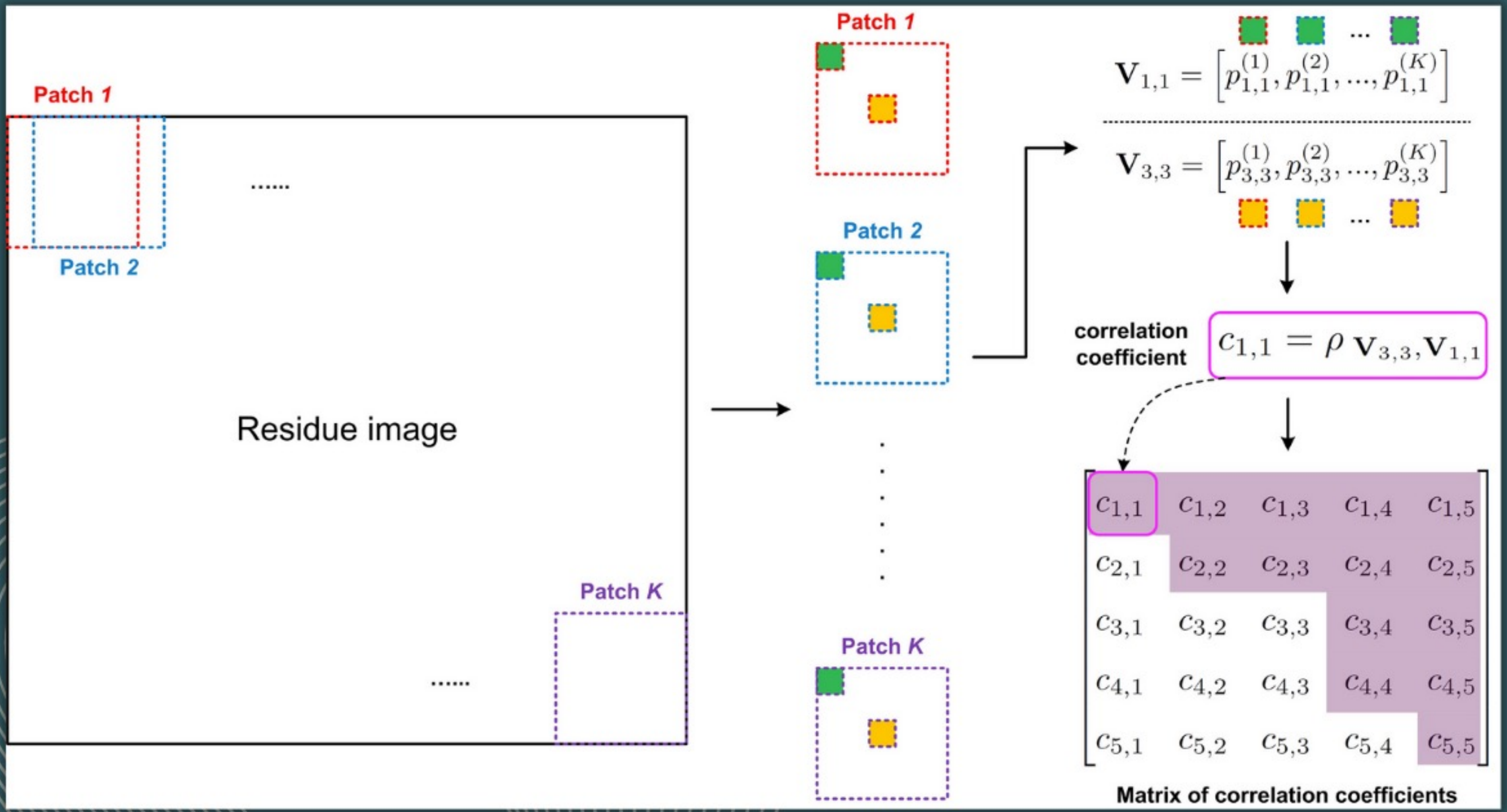
Patch 1
Patch 2
Residue image
Patch K
Patch 1
Patch 2
Patch K
$\mathbf{V}_{1,1} = \left[p_{1,1}^{(1)}, p_{1,1}^{(2)}, ..., p_{1,1}^{(K)}\right]$
$\mathbf{V}_{3,3} = \left[p_{3,3}^{(1)}, p_{3,3}^{(2)}, ..., p_{3,3}^{(K)}\right]$



$$c_{\mathrm{i,j}} = \frac{\sum_{k=1}^{K} (p^{(\mathrm{k})}{}_{3,\ 3} - \overline{p}_{3,\ 3})(p^{(\mathrm{k})}{}_{\mathrm{i,\ j}} - \overline{p}_{\mathrm{i,\ j}})}{\sqrt{\sum_{k=1}^{K} (p^{(\mathrm{k})}{}_{3,\ 3} - \overline{p}_{3,\ 3})^2}\sqrt{\sum_{k=1}^{K} (p^{(\mathrm{k})}{}_{\mathrm{i,\ j}} - \overline{p}_{\mathrm{i,\ j}})^2}}$$



$$c_{\mathrm{i,j}} = \frac{\sum_{k=1}^{K} ({p^{(\mathrm{k})}}_{3,\,3} - \overline{p}_{3,\,3})({p^{(\mathrm{k})}}_{\mathrm{i,\,j}} - \overline{p}_{\mathrm{i,\,j}})}{\sqrt{\sum_{k=1}^{K} ({p^{(\mathrm{k})}}_{3,\,3} - \overline{p}_{3,\,3})^2}\sqrt{\sum_{k=1}^{K} ({p^{(\mathrm{k})}}_{\mathrm{i,\,j}} - \overline{p}_{\mathrm{i,\,j}})^2}}$$



$$c_{\mathrm{i,j}} = \frac{\sum_{k=1}^{K} \left(p^{(\mathrm{k})}{}_{3,\,3} - \overline{p}_{3,\,3}\right)\left(p^{(\mathrm{k})}{}_{\mathrm{i,\,j}} - \overline{p}_{\mathrm{i,\,j}}\right)}{\sqrt{\sum_{k=1}^{K} \left(p^{(\mathrm{k})}{}_{3,\,3} - \overline{p}_{3,\,3}\right)^2}\sqrt{\sum_{k=1}^{K} \left(p^{(\mathrm{k})}{}_{\mathrm{i,\,j}} - \overline{p}_{\mathrm{i,\,j}}\right)^2}}$$

The CC values measure how the horizontal or vertical residues are related to each other through second-order mixed statistical moments.

The extracted CC values are sensitive to the alterations due to image recapture and exhibit different characteristics for single captured and recaptured images.

The mean CC values between the central pixel and most neighbors in the local 5x5 patch are noticeably higher for recaptured images than those for single captured images, in particular for the elements in the second, third and fourth columns of the two matrices. This increase is probably due to the introduced blurriness effect.

$$\overline{CC}_{recapture} = \begin{bmatrix} -0.2458 & 0.2078 & 0.5845 & 0.2160 & -0.2357 \\ -0.3344 & 0.2426 & 0.8129 & 0.2506 & -0.3295 \\ -0.3821 & 0.2239 & 1. & 0.2239 & -0.3822 \\ -0.3295 & 0.2506 & 0.8130 & 0.2426 & -0.3344 \\ -0.2357 & 0.2160 & 0.5846 & 0.2079 & -0.2459 \end{bmatrix}$$

$$\vee$$

$$\overline{CC}_{single} = \begin{bmatrix} -0.1697 & 0.0067 & 0.3742 & 0.0021 & -0.1688 \\ -0.3103 & -0.0329 & 0.6965 & -0.0339 & -0.3110 \\ -0.4233 & -0.0975 & 1. & -0.0975 & -0.4233 \\ -0.3110 & -0.0339 & 0.6965 & -0.0329 & -0.3103 \\ -0.1687 & 0.0021 & 0.3741 & 0.0067 & -0.1697 \end{bmatrix}$$

$$\overline{CC}_{recapture} = \begin{bmatrix} \boxed{-0.2458} & 0.2078 & 0.5845 & 0.2160 & \boxed{-0.2357} \\ -0.3344 & 0.2426 & 0.8129 & 0.2506 & -0.3295 \\ -0.3821 & 0.2239 & 1. & 0.2239 & -0.3822 \\ -0.3295 & 0.2506 & 0.8130 & 0.2426 & -0.3344 \\ \boxed{-0.2357} & 0.2160 & 0.5846 & 0.2079 & \boxed{-0.2459} \end{bmatrix}$$

$$\wedge$$

$$\overline{CC}_{single} = \begin{bmatrix} \boxed{-0.1697} & 0.0067 & 0.3742 & 0.0021 & \boxed{-0.1688} \\ -0.3103 & -0.0329 & 0.6965 & -0.0339 & -0.3110 \\ -0.4233 & -0.0975 & 1. & -0.0975 & -0.4233 \\ -0.3110 & -0.0339 & 0.6965 & -0.0329 & -0.3103 \\ \boxed{-0.1687} & 0.0021 & 0.3741 & 0.0067 & \boxed{-0.1697} \end{bmatrix}$$

The **mean CC values** between the central pixel and distant neighbors, in particular those at the four corners of the CC value matrix,
in **recaptured images** are **lower** than those in single captured images.
This decrease is probably caused by the **aliasing-like distortion**.

Pipeline of the adopted method

1

Pre-processing

Conversion to B&W and resizing of the image (width of 2048 pixels).

2

Computation of the residue images

3

Pixel-wise correlation coefficients

4

Feature Vector

Removing of redundancy and concatenation.

5

Classification

$$\text{CC of R1} = \begin{bmatrix} c1,1 & c1,2 & c1,3 & c1,4 & c1,5 \\ c2,1 & c2,2 & c2,3 & c2,4 & c1,5 \\ c3,1 & c3,2 & c3,3 & c3,4 & c3,5 \\ c4,1 & c4,2 & c4,3 & c4,4 & c4,5 \\ c5,1 & c5,2 & c5,3 & c5,4 & c5,5 \end{bmatrix}$$

$$\text{CC of R2} = \begin{bmatrix} c1,1 & c1,2 & c1,3 & c1,4 & c1,5 \\ c2,1 & c2,2 & c2,3 & c2,4 & c1,5 \\ c3,1 & c3,2 & c3,3 & c3,4 & c3,5 \\ c4,1 & c4,2 & c4,3 & c4,4 & c4,5 \\ c5,1 & c5,2 & c5,3 & c5,4 & c5,5 \end{bmatrix}$$

$$\text{CC of R1} = \begin{bmatrix} c1,1 & c1,2 & c1,3 & c1,4 & c1,5 \\ c2,1 & c2,2 & c2,3 & c2,4 & c1,5 \\ c3,1 & c3,2 & c3,3 & c3,4 & c3,5 \\ c4,1 & c4,2 & c4,3 & c4,4 & c4,5 \\ c5,1 & c5,2 & c5,3 & c5,4 & c5,5 \end{bmatrix}$$

$$\text{CC of R2} = \begin{bmatrix} c1,1 & c1,2 & c1,3 & c1,4 & c1,5 \\ c2,1 & c2,2 & c2,3 & c2,4 & c1,5 \\ c3,1 & c3,2 & c3,3 & c3,4 & c3,5 \\ c4,1 & c4,2 & c4,3 & c4,4 & c4,5 \\ c5,1 & c5,2 & c5,3 & c5,4 & c5,5 \end{bmatrix}$$

$$\text{CC of R1} = \begin{bmatrix} c1,1 & c1,2 & c1,3 & c1,4 & c1,5 \\ c2,1 & c2,2 & c2,3 & c2,4 & c1,5 \\ c3,1 & c3,2 & c3,3 & c3,4 & c3,5 \\ c4,1 & c4,2 & c4,3 & c4,4 & c4,5 \\ c5,1 & c5,2 & c5,3 & c5,4 & c5,5 \end{bmatrix}$$

$$\text{CC of R2} = \begin{bmatrix} c1,1 & c1,2 & c1,3 & c1,4 & c1,5 \\ c2,1 & c2,2 & c2,3 & c2,4 & c1,5 \\ c3,1 & c3,2 & c3,3 & c3,4 & c3,5 \\ c4,1 & c4,2 & c4,3 & c4,4 & c4,5 \\ c5,1 & c5,2 & c5,3 & c5,4 & c5,5 \end{bmatrix}$$

Final feature vector = [c1,1 c1,2 c1,3 c1,4 c1,5 c2,2 c2,3 c2,4 c1,5 c3,4 c3,5 c4,4 c4,5 c5,5 c1,1 c1,2 c1,3 c1,4 c1,5 c2,2 c2,3 c2,4 c1,5 c3,4 c3,5 c4,4 c4,5 c5,5]

Pipeline of the adopted method

1

Pre-processing

Conversion to B&W and resizing of the image (width of 2048 pixels).

2

Computation of the residue images

3

Pixel-wise correlation coefficients

4

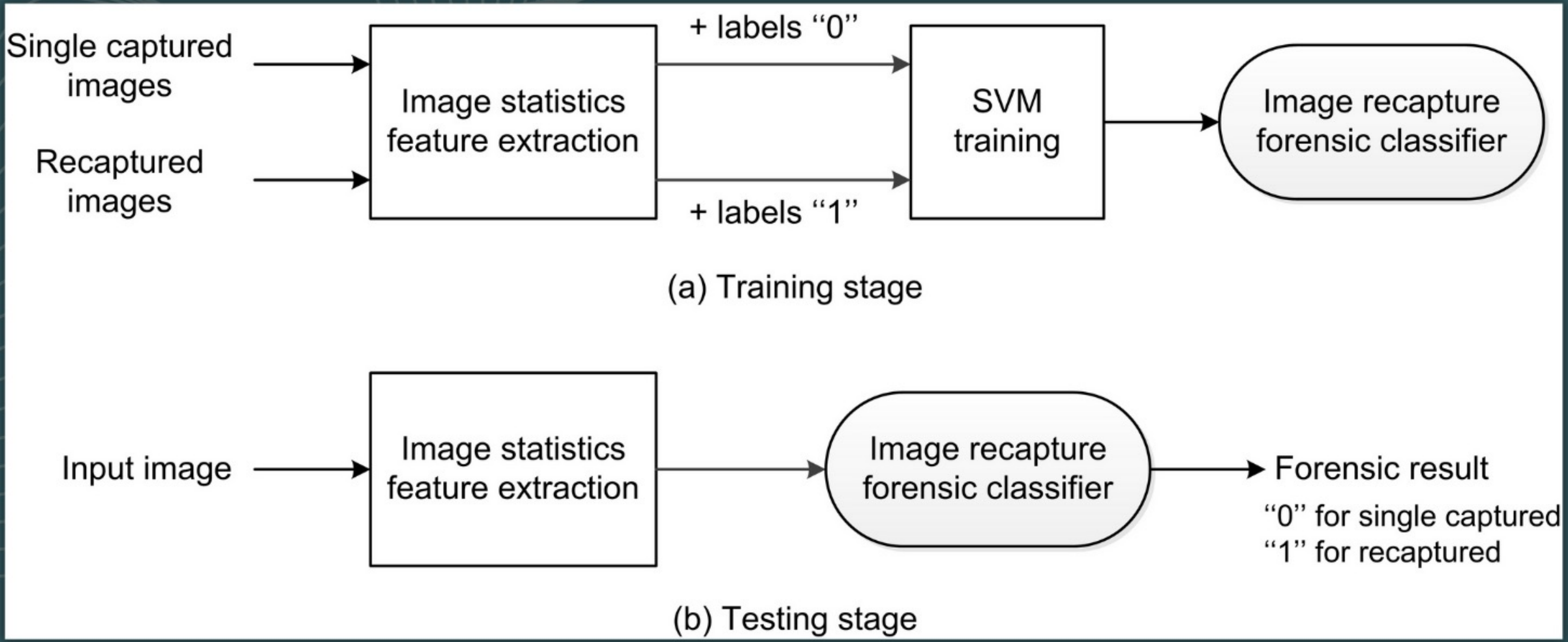
Feature Vector

Removing of redundancy and concatenation.

5

Classification

Classification



Classification

15:100

The ratio between the training and the testing sets.

01

Training

02

Testing

Classification

SVM classifier

kernel RBF

01

Training

02

Testing

Classification

Hyperparameters determined with a 5-fold cross-validation performing a grid-search on predefined sets of values by maximizing the accuracy values.

01

Training

02

Testing

The values of the hyperparameters have been determined referring to:

Chih-Wei Hsu, Chih-Chung Chang, Chih-Jen Lin, 2016,
A Practical Guide to Support Vector Classification,
Department of Computer Science, National Taiwan University.

Classification

C : [2^(–5), ..., 2^(15)]

gamma : [2^(-15), ..., 2^3]

01

Training

02

Testing

The values of the hyperparameters have been determined referring to:

Chih-Wei Hsu, Chih-Chung Chang, Chih-Jen Lin, 2016,
A Practical Guide to Support Vector Classification,
Department of Computer Science, National Taiwan University.

Classification

50 runs of cross-validation and testing

01

Training

02

Testing

Classification

50 runs of cross-validation and testing

for each run

- Overall accuracy
- Single-captured accuracy
- Recaptured accuracy

01

Training

02

Testing

Classification

50 runs of cross-validation and testing

for each run

- Overall accuracy
- Single-captured accuracy
- Recaptured accuracy

Final values of accuracies determined as the means.

01

Training

02

Testing

Implementation choices

ICL data-set

VS

ROSE data-set

Implementation choices

Handling of the ICL data-set

- The ICL data-set is composed of 900 single captured and 1440 recaptured images.

Implementation choices

Handling of the ICL data-set

- The ICL data-set is composed of 900 single captured and 1440 recaptured images.
- We created a folder named AllImages containing all the images of the data-set.
- We created labels_final.txt which defines the type of each image.

Implementation choices

Handling of the ICL data-set

- The ICL data-set is **composed** of 900 single captured and 1440 recaptured images.
- We created a folder named **AllImages** containing all the images of the data-set.
- We created **labels_final.txt** which defines the type of each image.
- *Y* is the vector of the **labels** (0 = Original Captured,1 = Recaptured).
- X is a matrix of dimension [n_images×28] containing for each image its **feature vector**.

Implementation choices

Computation of the CC

Each correlation coefficient is computed using the *scipy* command *pearsonr*. In fact, the direct implementation through the formula takes too long to be computed.

Implementation choices

Implementation of the classifier

The classifier is implemented using the *sklearn* libraries.
In particular, we used the commands:

- *GridSearchCV* to define the classifier
- *accuracy_score* to compute the overall accuracy
- *recall_score* to compute the single-captured and recaptured accuriacies.

Implementation choices

Decision of the rescaling width

The paper reports the results of applying the described method to different image widths. We chose to implement the 2048 pixels case, as it seemed the most significant one for comparison purposes.

Implementation choices

Selection of the patch's dimension

We opted for 5×5 pixels as the size for the overlapping patches because the paper shows that it gives the best results compared to 7×7 pixels and 3×3 pixels patches within the chosen width of 2048 pixels.

94.85% Overall accuracy over 100 images

The process is applied on a simplified data-set, composed only by 100 images. This data-set is available on the same link of the ICL data-set and it is composed by 50 single captured images and 50 recaptured ones.

Overall accuracy over the ICL data-set 97.34%

94.85% Overall accuracy over 100 images

The process is applied on a simplified data-set, composed only by 100 images. This data-set is available on the same link of the ICL data-set and it is composed by 50 single captured images and 50 recaptured ones.

Overall accuracy over the ICL data-set 97.34%

We can notice that the overall accuracy of the reduced data-set is quite lower than to the one of the complete data-set.

For what concerns the accuracies obtained in the single iterations of the cycle, in the complete case these values are steady around their final mean value, instead for the *reduced case* they vary in a *wide range*.

In the reduced data-set the proportion 15:100 of the number of the training images and the number of testing images is very unbalanced, so the training phase is less reliable.

Comparison with the paper results

Method	Overall accuracy	Original captured accuracy	Recaptured accuracy
Paper	97.71%	96.27%	98.62%
Ours	97.34%	96.23%	98.04%

Remarks

Comparison with the paper results

Method	Overall accuracy	Original captured accuracy	Recaptured accuracy
Paper	97.71%	96.27%	98.62%
Ours	97.34%	96.23%	98.04%

Our accuracy values are slightly lower than the ones reported in the paper: since the author has not specified a method to determine the hyperparameters in the cross validation, he might have adopted a different approach from us.

The accuracies of recaptured images are quite higher than the ones of single captured images. This is probably due to the higher number of recaptured images present in the data-set with respect to the number of single captured images.

Final considerations

Although it is quite simple as it is image-statistics-based, the method proposed by the author appears to be discriminative enough to well detect the differences between single captured and recaptured images.

It achieves slightly higher classification accuracy when compared with other methods from which it is different due to the fact that the current trend is the one of developing more sophisticated recapture forensic methods by either combining multiple features or using complex machine learning tools.

Thanks for your attention