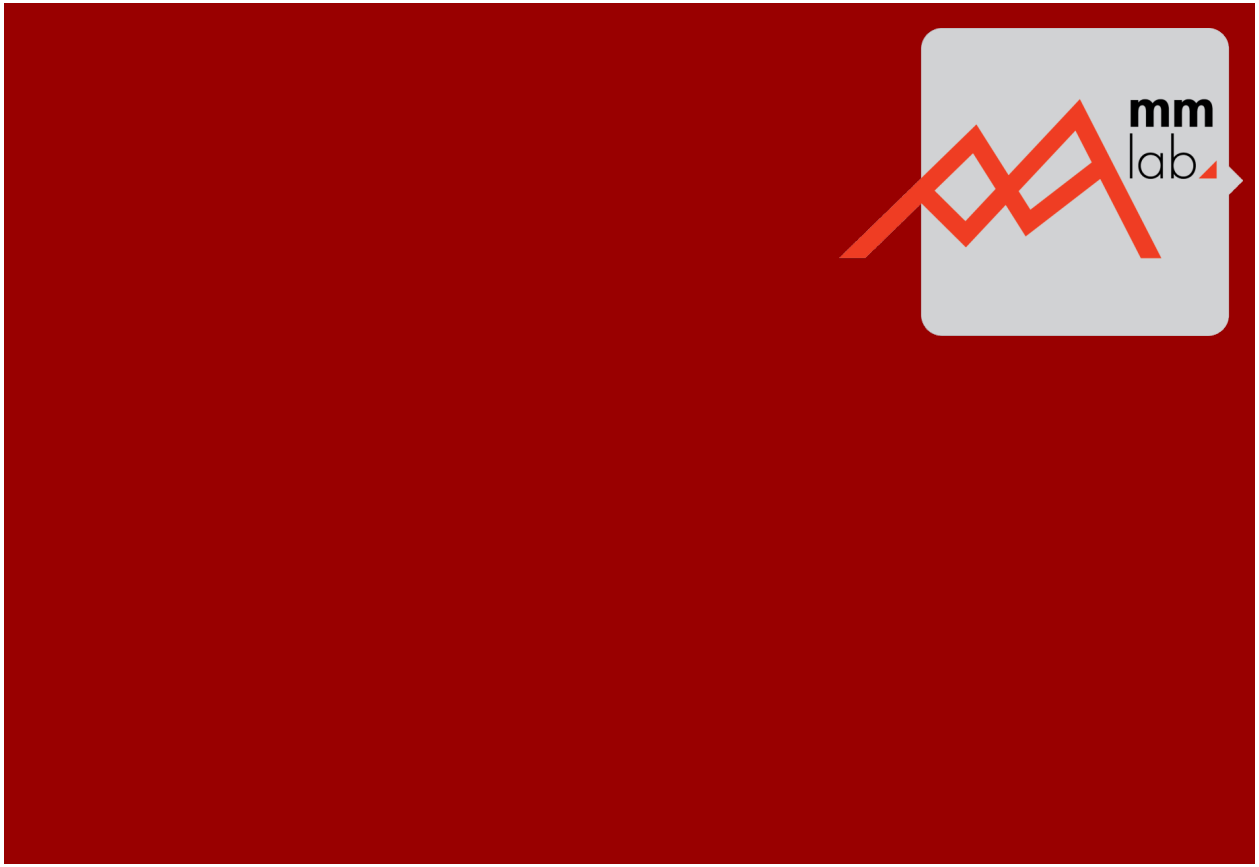


Image Forensics
Source Identification

Multimedia forensics - what

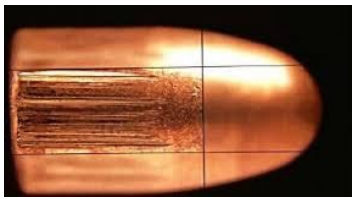
Identify the device that took a particular multimedia data

Determine possible tampering

Discriminate multimedia objects generated by real cameras, scanners, or via CG softwares

Source identification

Every time a device acquires a multimedia object it leaves imperceptible traces on it

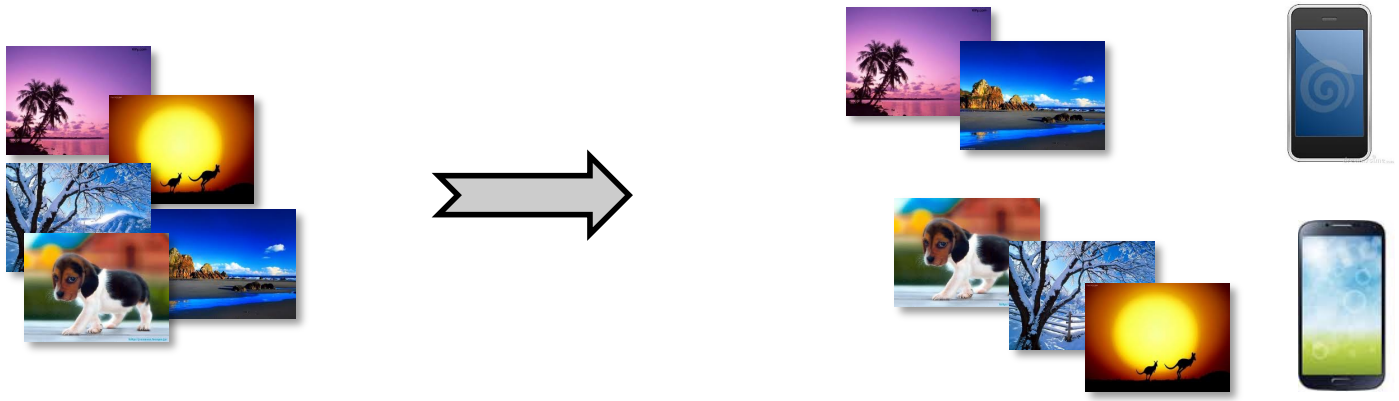


Source identification

- The aim is to identify the device which captured the content.
- Exploit traces left by the different steps taken during the image acquisition process.
- The basic idea comes from classical forensic science, where bullet analysis is carried on based on the distinct markings introduced onto it when fired. Such markings are distinctive and unique for every specific gun and can be therefore used to link the fired bullet with the weapon which shot it. Similarly, when shooting an image, a specific and unique fingerprint is introduced into the content, depending on the device which took it.

Source identification

- In general, **source identification** problem links multimedia content to a particular (class of) acquisition device(s).



- It can be solved at different levels …

Source identification

- **Level 1**
  - Which Device ?

Which Coolpix P60?

Source identification

- **Level 2**
  - Which BRAND/MODEL?

Nikon?
Canon?
Sony?


Nikon D50?
Nikon D7000?


Canon EOS 5D?
Canon Powershot G3X?

Sony Cybershot RX100?
Sony Alpha a6500?


Source identification

- **Level 3**
  - Which CLASS of device?

Digital Camera

Scanner

Computer Generated

Smartphone

Source identification

- Source model Identification / Individual Source Identification
- Exploit traces left by the different steps taken during the image acquisition process, in particular the artifacts introduced by camera as light passes through its optics and is registered by the sensor.

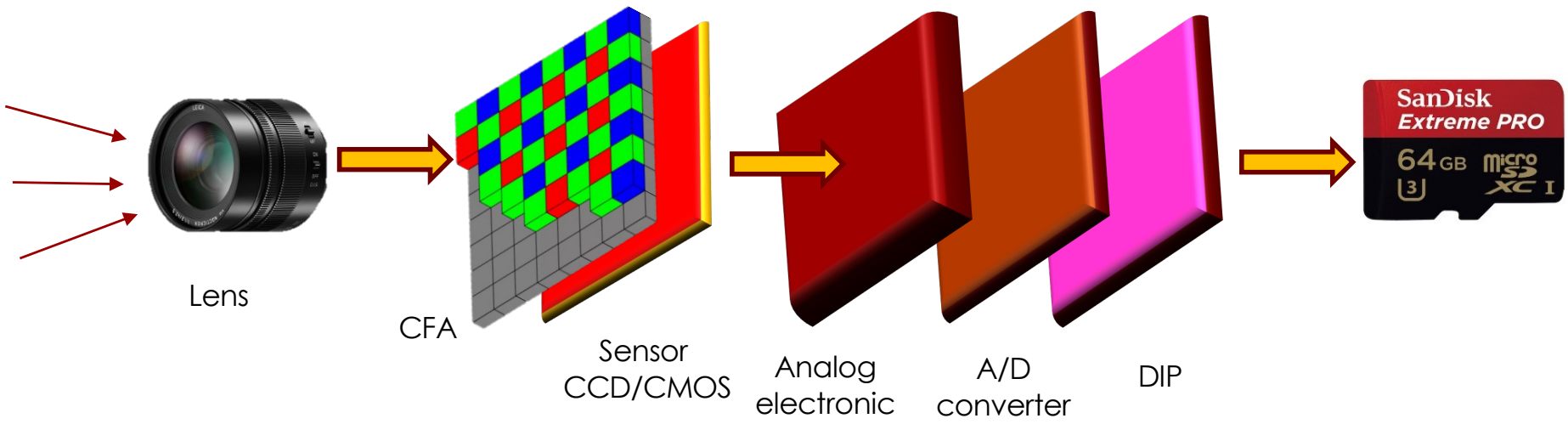


Image Pipeline Acquisition

- Lens system
  - Composed of a lens and the mechanisms to control exposure, focusing, and image stabilization to collect and control the light from the scene.
- Filters
  - Includes the infra-red and anti-aliasing filters to ensure maximum visible quality.
- Image sensor
  - An image sensor is an array of rows and columns of photodiode elements, or pixels. When light strikes the pixel array, each pixel generates an analog signal proportional to the intensity of light, which is then converted to digital signal and processed by the DIP.

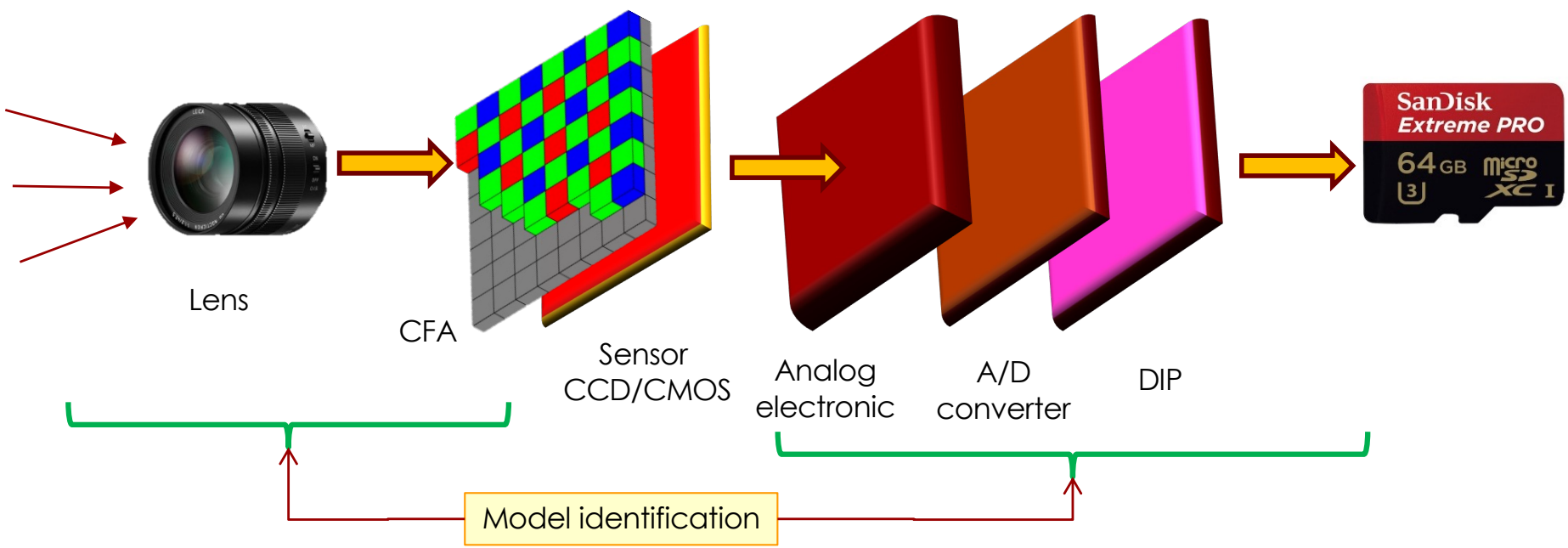
Image Pipeline Acquisition

- CFA
  - Since the sensor pixels are not sensitive to color, to produce a color image, a color filter array (CFA) is used in front of the sensor so that each pixel records the light intensity for a single color only.
- We are able to distinguish some 10 million colors from just three wavelength measurements, each made by a different type of cone: L-cone (long – orange and red), M-cone (medium – green and yellow), S-cone (short – violet and blue).
- Analogous to the eye's three cone types, digital cameras have three channels with peak sensitivities at different wavelengths: R, G, B. This means that each pixel is represented as three values.

Image Pipeline Acquisition

- DIP
  - The output from the sensor with a Bayer (RGB) filter (assume) is a mosaic of red, green and blue pixels of different intensities. Each pixel contains the information of only one color. The digital image processor implements interpolation (demosaicing) algorithms to recover the missing information of the other two colors for each pixel.
  - The Bayer pattern has twice as many G filters as R or B filters because it is designed to mimic the human retina which is most sensitive to light in the green range of the spectrum. Unlike the human retina, R-G-B filters of the CFA are distributed in periodic pattern.
  - The DIP also performs further processing such as white balancing, noise reduction, matrix manipulation, image sharpening, aperture correction, and gamma correction to produce a good quality image.

Source identification - device



- Sensor CCD/CMOS presents intrinsic imperfection
- It leaves a systematic noise on each acquired image (video)
- Such noise is imperceptible

Source identification - device

- Such noise has two components

Fixed Pattern Noise (FPN)

Dark current (usually compensated directly in the device)

Photo Response Non-Uniformity Noise (PRNU)

Not uniform response to the same light impulse

Sensor Imperfection

- In a ideal, noise-free system, the amount of recorded light would be directly proportional to the pixel values in the final digital image. In reality, there are a variety of factors that introduce discrepancies between the amount of light that is initially recorded and the final digitized pixel values.
- Matching the source by identifying and extracting systematic errors due to imaging sensor, which reveal themselves on all images acquired by the sensor in a way independent of the scene content.
- Sensor's pixel defects and pattern noise (fixed pattern noise + photo response non-uniformity noise).
- Detect traces of defective pixels, such as hot pixels, dead pixels, pixel traps, cluster defects.

Sensor Noise

- Each pixel in a digital camera's sensor records the amount of incident light that strikes it. Slight imperfections in manufacturing introduce small amounts of noise in the recorded image.
- This noise is spatially varying and consistent over time and can therefore be used for forensics and ballistic purposes.
- The image imperfections can be modeled as:

$$I(x,y) = I_0(x,y)+\gamma I_0(x,y)K(x,y)+N(x,y)$$

where $I_0(\cdot)$ is the noise-free image, γ is a multiplicative constant, $K(\cdot)$ is the multiplicative noise (termed photo-response non-uniformity noise (PRNU)), and $N(\cdot)$ is an additive noise term. The multiplicative PRNU factor is used for forensic and ballistic purposes.

Sensor Noise

- This noise arises from slight variations in the size and material properties of the sensor cells themselves. Physical inconsistencies across the sensor cells lead to differences in the efficiency with which the cells convert light into digital pixel values.
- Some cells consistently under-report the amount of measured light, while others consistently over-report the amount of measured light. These variations, termed photo-response non-uniformity, lead to a stable noise pattern that is distinctive to the device.
- PRNU modulates the pixel proportional to its value. It is a fixed property of the sensor and it does not vary from image to image.

Sensor Noise

- The PRNU is estimated from a series of authentic images taken from the camera in question. Each image is denoised with any standard denoising filter and subtracted from the original image

$$W_k(x,y) = I_k(x,y) - I_k\hat{}(x,y)$$

where $I_k\hat{}(x,y)$ are the denoised images (k=1,…,N).

The terms $W_k(x,y)$ suppress the underlying image content and make the estimation of the PRNU more reliable. The PRNU is estimated as

$$K(x,y)=\text{sum } W_k(x,y)I_k(x,y)/\text{sum } I_k^2(x,y)$$

Sensor Noise

- The PRNU can then be used to determine if an image originated from a specific camera, or if a portion of an image has been altered.

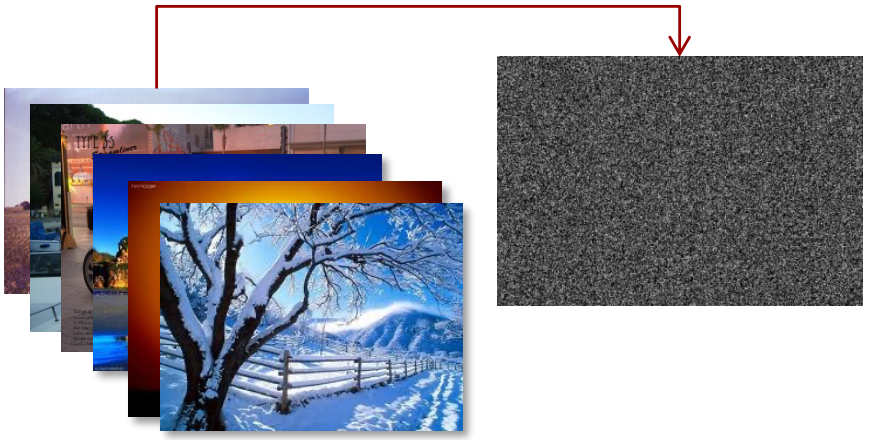
- For the latter application, an image in question I(x,y) is denoised and subtracted from itself to yield W(x,y) as described above. The PRNU K(x,y) is estimated from a set of images known to have originated from the same camera as I(x,y). The correlation between the PRNU and the image being analyzed is given by

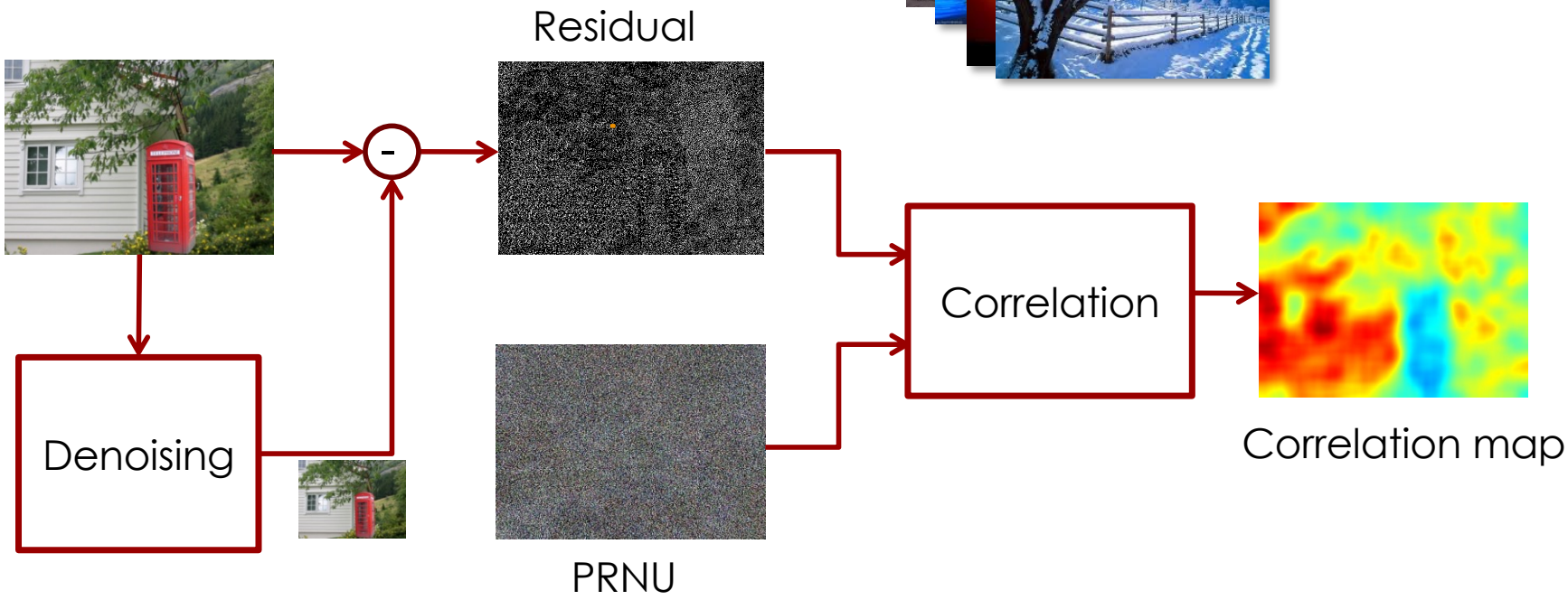
$$\rho = I(x, y)K(x, y) \otimes W(x, y)$$

where $\otimes$ denotes normalized correlation. The correlation ρ is used as a measure of authenticity and can be computed locally in order to detect localized tampering.

Sensor Noise

- **Photo Response Non Uniformity Noise** PRNU estimation and detection





Sensor Noise

- The best images for estimation of PRNU are those with high luminance (but not saturated) and smooth content. If the camera under investigation is in our possession, out-of-focus images of bright cloudy sky would be the best.
- In practice good estimates of the fingerprint may be obtained from 20-50 natural images depending on the camera.
- For color images the PRNU will be highly correlated across the RGB color channels. There is little advantage to computing the PRNU for each color channel. The RGB image can be converted to a single gray-scale image from which the PRNU is estimated.

Sensor Noise

- Different denoising filters can impact on the final performance.
- Performance may benefit from the use of more sophisticated filters.
- Moreover, sophisticated filters allow decreasing the number of picture necessary for the reference pattern definition.

V. Conotter and G. Boato, "Analysis of sensor fingerprint for source camera identification", Electronics Letters, vol. 47, n. 25, 2011.

Sensor Noise

- PRNU can be used for a variety of digital forensics tasks
  - Device identification
  - Device linking (prove that two images were taken by the same device)
  - Recovery of processing history (presence of camera fingerprint indicates that is natural and not a computer rendering – the strength or form of the fingerprint can indicate particular processing)
  - Detection of digital forgeries (absence of the fingerprint in individual image regions)
  - Blind source clustering

- It is essentially an unintentional stochastic spread-spectrum watermark (it survives lossy compression, filtering, gamma correction, and many other typical processing).

Sensor Noise

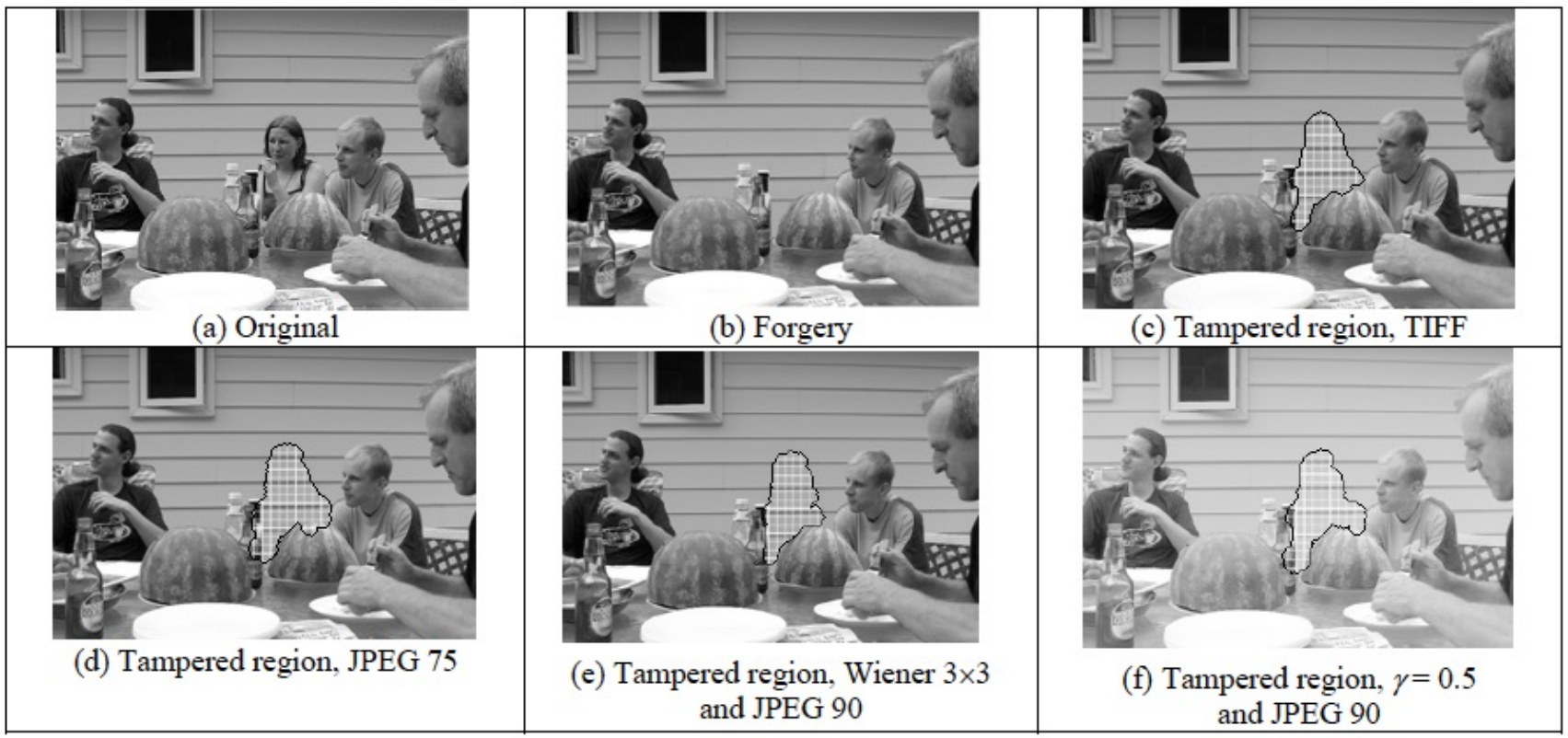
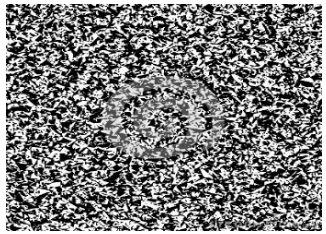


Fig. 8: An original (a) and forged (b) Olympus C765 image and its detection from a forgery stored as TIFF (c), JPEG 75 (d), denoised using a 3×3 Wiener filter and saved as 90% quality JPEG (e), gamma corrected with $\gamma = 0.5$ and stored as 90% quality JPEG.

J. Fridrich "Digital Forensics Using Sensor Noise", IEEE Signal Processing Magazine, 2009.

Sensor Noise

- **PRNU based device identification**



Open issues

- **Robustness**
  - JPEG compression
  - Different resolutions
  - Various processing
- **Security**
  - Fingerprint cancellation or replacement
  - Image quality preserved
- **Computational complexity**
  - Dataset of fingerprints
  - Fingerprint digest

Source identification - model

- **Level 2**
  - Which BRAND/MODEL?

Nikon?
Canon?
Sony?

Nikon D50?
Nikon D7000?

Canon EOS 5D?
Canon Powershot G3X?

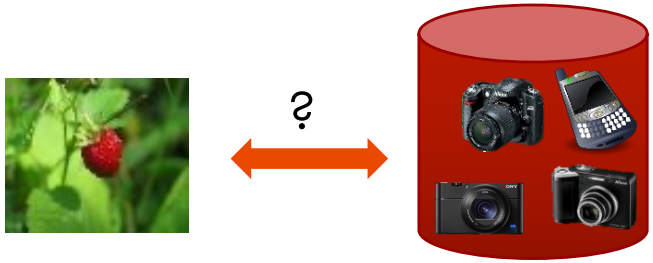
Sony Cybershot RX100?
Sony Alpha a6500?

Source identification - model

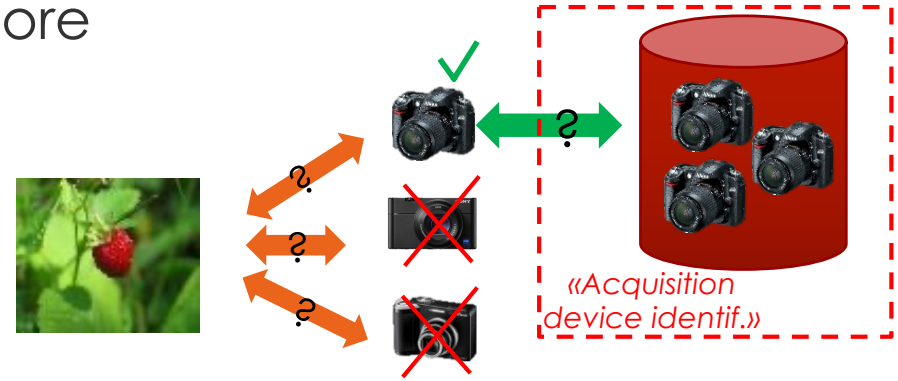
- Several applications:

1. Check compatibility w.r.t an alleged provenance device (even if it's not available!)

2. Trace back a picture to the kind of device that captured it



3. Pre-filtering step before more specific analyses



Source identification - model

Detection among different models

Detection among different devices

Source identification - model

instances of these traces are specific to the respective (class of) device(s)

real word scene

lenses

optical filter

CFA pattern

CCD or CMOS sensor

imaging sensor

CFA interpol ation

in-camera software processing (white balancing; contrast saturation; etc)

in-camera compression & coding

digital image

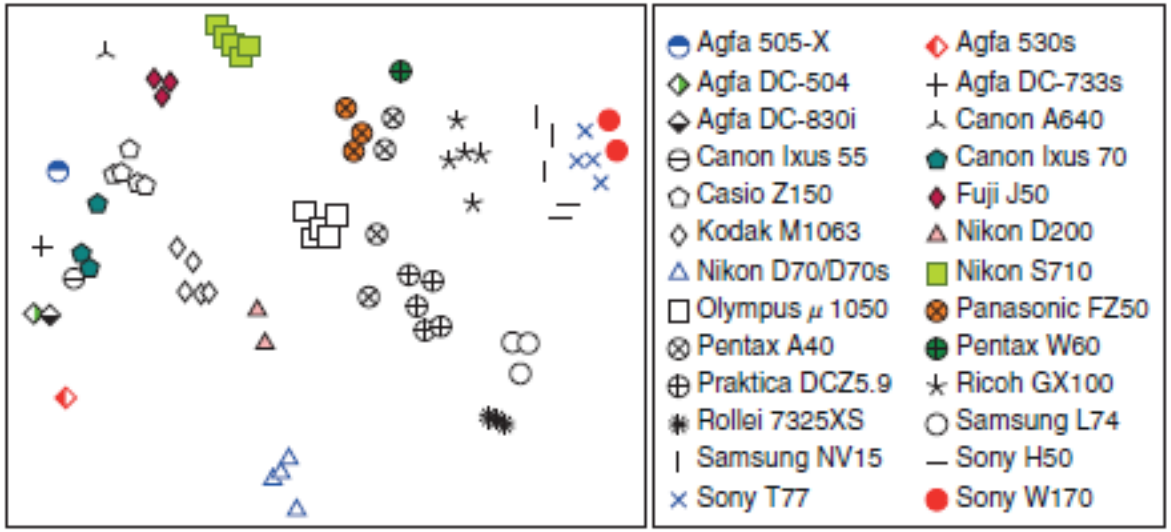
out-camera processing (Photoshop, social network sharing,…)

final digital image

- Each phase leaves distinctive footprints!
  - ✓ at the signal level
  - ✓ at the metadata/file container level

Scenarios - Perfect knowledge

- Finite set of known camera models
  - for each model we have enough training images to carry out reliable estimates of features of interest
- We can use a multiclass classifier and will be able to compute a full confusion matrix



Scenarios - Limited knowledge

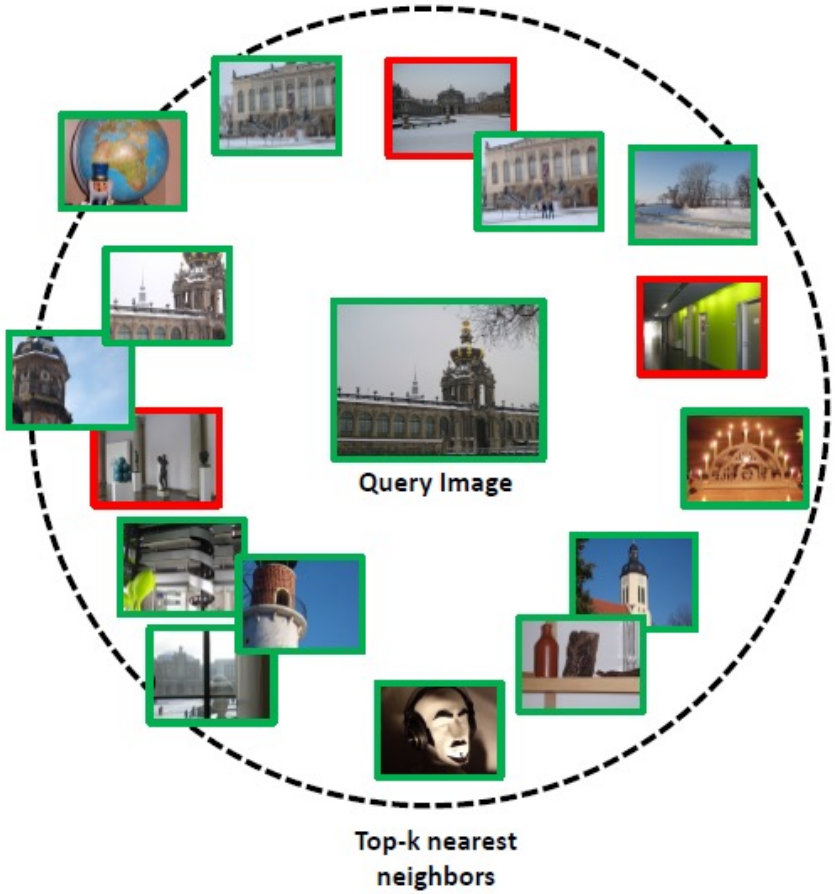
- Full knowledge on target model, but know nothing about the number of other models
- Practically, we have a large number of training images, classified as either belonging to the target model or another (unknown) model
  - Here, we can use only a one-class classifier

Other models

Query image model

Scenarios - Zero knowledge

- No prior information on the number of camera models
- There are only a large number of images
- We can only retrieve other images taken from the same model, which may help for subsequent investigations



Q.T. Phan, G. Boato, F. De Natale. Accurate and Scalable Image Clustering Based on Sparse Representation of Camera Fingerprint. IEEE Transactions on Information Forensics & Security, 2018

Source identification - model

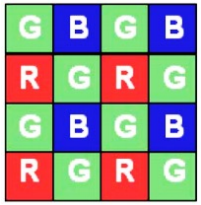
Signal level

- Artifacts related to specific in-camera processing steps:
  - Lens, CFA, In-camera processing
- General intrinsic features, irrespective of their physical meaning (deep learning)

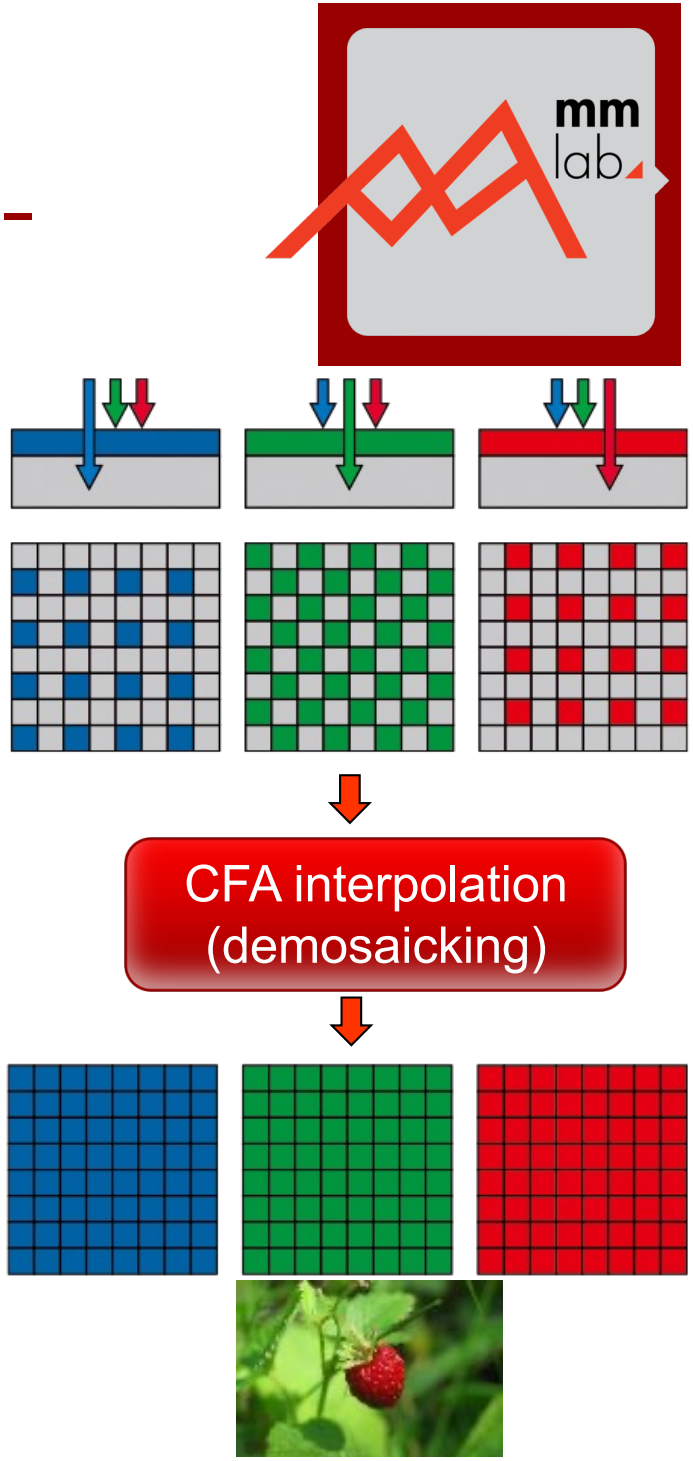
Format level

- Codec
- Metadata
- File container

Signal-level Traces: CFA-based

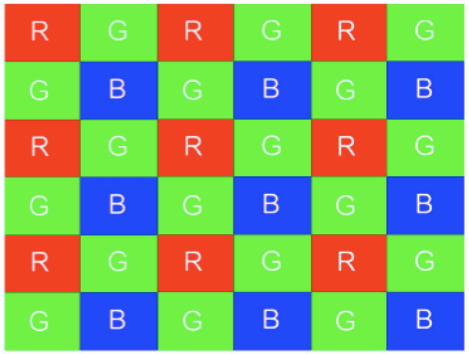


- Colour Filter Array
  - A thin film that selectively permits a certain component of light to pass through it to the sensor.
  - In practice, to each pixel only one particular colour is gathered.
- The sensor output is successively interpolated to obtain all the 3 colours for each pixel (demosaicking)



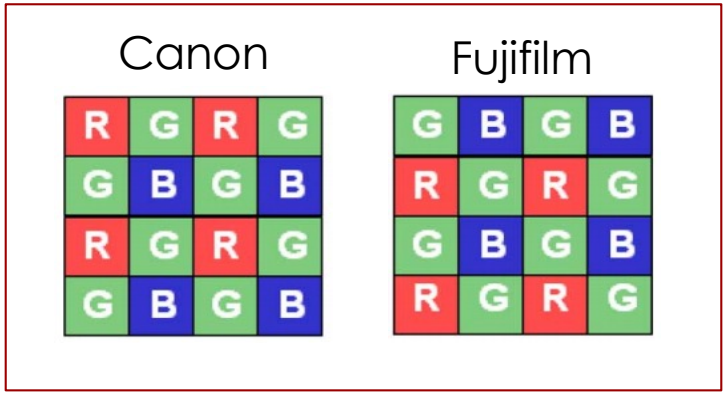
Signal-level Traces: CFA-based

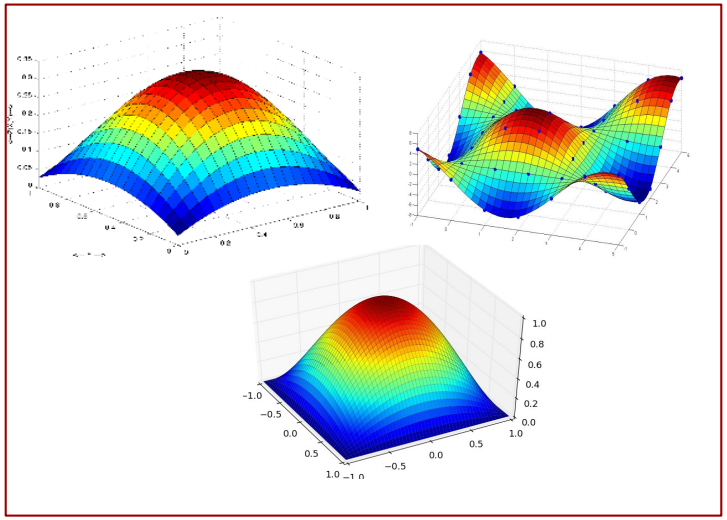
- The interpolation algorithm, used to estimate the missing color values, introduces correlation between neighbor pixels. Since CFA patterns are typically periodic these correlations will be periodic as well.
- Assume that each pixel value is correlated to its neighbors with the associated weighting coefficients and each camera manufacturer uses different interpolation kernel and/or different weighting coefficients.
- The crux of this approach lies on estimating these coefficients and associating them with the digital camera-model used to capture the images.



Signal-level Traces: CFA-based

- These artifacts are inherent to camera manufacturing processes and inconsistencies can also be taken as evidence of tampering.

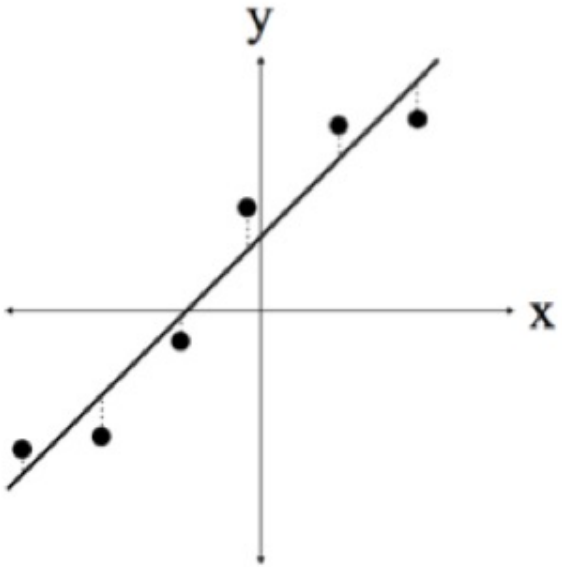




Least squares

- Fit a line to the data (x_i,y_i) i=1,…,n
- Model of a line y=mx+b
- Over-constrained problem X**u**=**y**

$$\begin{pmatrix} x_1 & 1 \\ x_2 & 1 \\ \dots & \dots \\ x_n & 1 \end{pmatrix} \begin{pmatrix} m \\ b \end{pmatrix} = \begin{pmatrix} y_1 \\ y_2 \\ \dots \\ y_n \end{pmatrix}$$



Least squares

- We seek a solution that minimizes some measure of goodness of fit of a line to the points.
- Minimize the overall vertical displacement mx+b-y of the points from the line.
- Minimize the sum of these squared distances: $E(\mathbf{u}) = ||X\mathbf{u}-\mathbf{y}||^2$
- Differentiating this error and setting it equal to zero we get:

  $dE/d\mathbf{u} = 2X^T(X\mathbf{u}-\mathbf{y}) = 0 \quad \Rightarrow \quad \mathbf{u} = (X^TX)^{-1}X^T\mathbf{y}$
- This basic framework is applicable to estimate any model that is linear in their unknown parameters.

Least squares

- Weighted Least Squares allows for a non-uniform treatment of the contribution of each individual point to the overall error

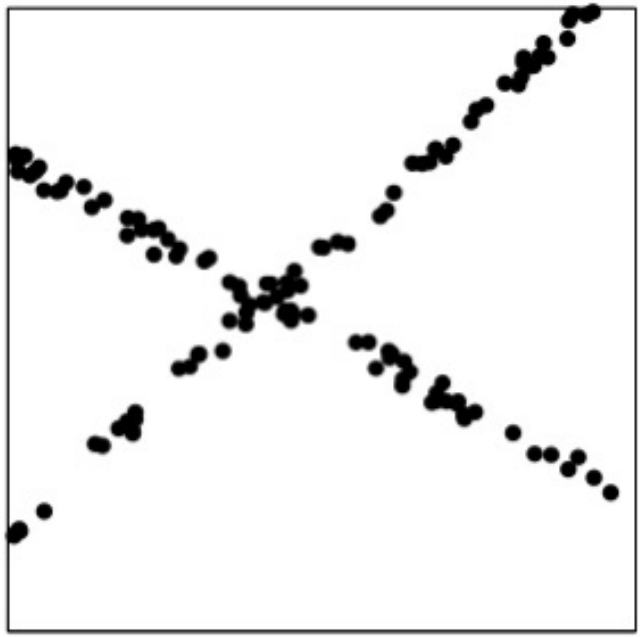
$$E(\mathbf{u}) = W \, || \, X\mathbf{u} - \mathbf{y} \, ||^2$$

where W is a diagonal weighting matrix with diagonal elements w_i corresponding to the weight associated with the i^{th} point.

- A larger weight places more emphasis on minimizing that point's deviation from the model.
- Notice that if W is the identity matrix then this solution reverts back to the classical least squares solution.

Expectation Maximization

- The Expectation/Maximization (EM) algorithm simultaneously segments and fits data generated from multiple parametric models.



$$y(i) = a_1 x(i) + b_1$$

$$y(i) = a_2 x(i) + b_2$$

Expectation Maximization

- Given model parameters, determine which data point was generated by which model:
  - Choose, for each data point i, the model *k* that minimizes the error between the data and the model prediction:

$$r_k(i) = |a_k x(i) + b_k - y(i)|$$

- Given which data points were generated by which model, estimate the model parameters:
  - Solve, for each model *k*, an over-constrained set of linear equations:

$$\begin{pmatrix} x_k(1) & 1 \\ x_k(2) & 1 \\ \vdots & \vdots \\ x_k(n) & 1 \end{pmatrix} \begin{pmatrix} a_k \\ b_k \end{pmatrix} = \begin{pmatrix} y_k(1) \\ y_k(2) \\ \vdots \\ y_k(n) \end{pmatrix}$$

Expectation Maximization

- What if we do not know anything (model assignment nor parameters)? The EM algorithm is an iterative two step algorithm that estimates both the model assignment and parameters.

- E – STEP
  - Assume that the model parameters are known;
  - Calculate the likelihood of each data point belonging to each model.

$$r_k(i) = a_k x(i) + b(k) - y(i)$$

$$P(a_k, b_k | r_k(i)) = \frac{P(r_k(i)|a_k, b_k) P(a_k, b_k)}{P(r_k(i))}$$

$$= \frac{P(r_k(i)|a_k, b_k)}{P(r_1(i)|a_k, b_k) + P(r_2(i)|a_k, b_k)}$$

Expectation Maximization

- Assume Gaussian probability distribution:

$$w_k(i) = P(a_k, b_k | r_k(i)) = \frac{e^{-r_k^2(i))/\sigma}}{e^{-r_1^2(i))/\sigma} + e^{-r_2^2(i))/\sigma}}$$

where σ is proportional to the amount of noise in the data.

Expectation Maximization

- M – STEP
  - Take the likelihood of each data point belonging to each model, and re-estimates the model parameters using weighted least-squares, that is the following error function on the model parameters is minimized:

$$E_k(a_k, b_k) = \sum_{i=1}^{n} w_k(i)[a_k x(i) + b_k - y(i)]^2.$$

  - Each data point contributes to the estimation of each model in proportion to the belief that it belongs to that particular model.
  - Partial derivatives with respect to parameters set to zero:

$$\begin{pmatrix} \sum_{i=1}^{n} w_k(i)x(i)^2 & \sum_{i=1}^{n} w_k(i)x(i) \\ \sum_{i=1}^{n} w_k(i)x(i) & \sum_{i=1}^{n} w_k(i) \end{pmatrix} \begin{pmatrix} a_k \\ b_k \end{pmatrix} = \begin{pmatrix} \sum_{i=1}^{n} w_k(i)x(i)y(i) \\ \sum_{i=1}^{n} w(i)y(i) \end{pmatrix}$$

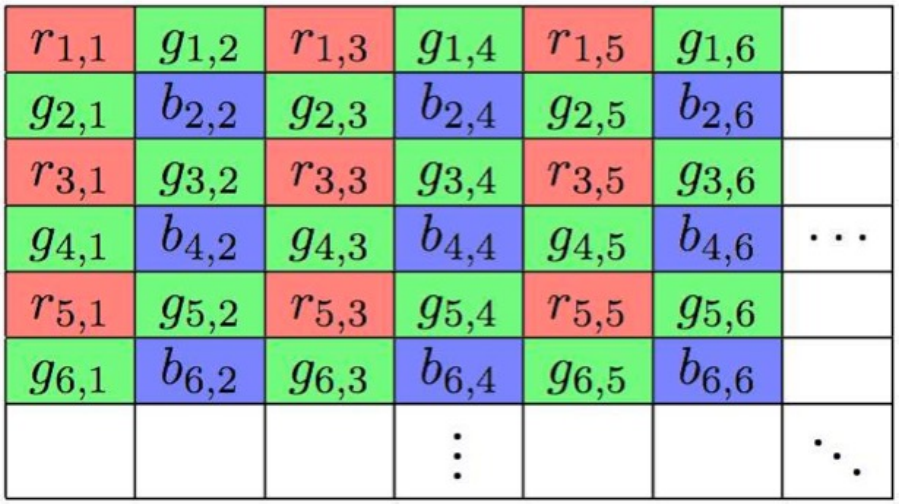
$$A\mathbf{x_k} = \mathbf{b}$$

Expectation Maximization

- The EM algorithm iteratively executes the “E” and “M” step, repeatedly estimating and refining the model assignments and parameters.
  - Initially the model parameters are randomly assigned.
  - The algorithm can be sensitive to the value of σ used.

Color Filter Array

- Only one third of the samples in a color image are captured by the camera, the other two thirds are interpolated.
- This interpolation introduces specific correlations between the samples of a color image.
- When creating a digital forgery these correlations may be destroyed or altered.



Color Filter Array

- Since a single color sample is recorded at each pixel location S(x,y), the other two color samples must be estimated from the neighboring samples in order to obtain a 3-channel color image R,G,B.

$$
\begin{aligned}
\tilde{R}(x,y) &= S(x,y) && \text{if } S(x,y) = r_{x,y} \\
&= 0 && \text{otherwise} \\
\tilde{G}(x,y) &= S(x,y) && \text{if } S(x,y) = g_{x,y} \\
&= 0 && \text{otherwise} \\
\tilde{B}(x,y) &= S(x,y) && \text{if } S(x,y) = b_{x,y} \\
&= 0 && \text{otherwise}
\end{aligned}
$$

CFA Interpolation

- The estimation of the missing color samples is referred to as CFA interpolation or demosaicking.
- The simplest methods for demosaicking are kernel-based interpolation methods that act on each channel independently.
- Linear filtering :

$$R(x,y) = \sum_{u,v=-N}^{N} h_r(u,v)\tilde{R}(x-u,y-v)$$

$$G(x,y) = \sum_{u,v=-N}^{N} h_g(u,v)\tilde{G}(x-u,y-v)$$

$$B(x,y) = \sum_{u,v=-N}^{N} h_b(u,v)\tilde{B}(x-u,y-v),$$

CFA Interpolation

- $h_r()$, $h_b()$, $h_g()$ are linear filters of size (2N+1)x(2N+1)
- Since the color filters in a CFA are typically arranged in a periodic pattern, these correlations are periodic.
- Consider the Bayer array where:

$$
\begin{aligned}
R(2x+1,2y) &= \frac{R(2x+1,2y-1)}{2} + \frac{R(2x+1,2y+1)}{2} \\
R(2x,2y+1) &= \frac{R(2x-1,2y+1)}{2} + \frac{R(2x+1,2y+1)}{2} \\
R(2x,2y) &= \frac{R(2x-1,2y-1)}{4} + \frac{R(2x-1,2y+1)}{4} \\
&+ \frac{R(2x+1,2y-1)}{4} + \frac{R(2x+1,2y+1)}{4}
\end{aligned}
$$

CFA Interpolation

- CFA interpolated images can be detected (in absence of noise) by noticing this correlation with neighbors.
- Non-interpolated samples are less likely to be correlated in the same manner.
- It is likely that tampering will destroy these correlations, or that splicing together two images from different cameras will create inconsistent correlations across the composite image.

CFA Interpolation

- Specific form of correlation
  - Simple linear model: easy to parametrize and reasonable approximations of CFA algorithms
- Know which samples are correlated to their neighbors
  - Ignore inter-channel correlations
- EM Algorithm
  - in the E-step the probability of each sample belonging to each model is estimated;
  - in the M-step the specific form of the correlations between samples is estimated.

CFA Interpolation

- E – STEP
  - Assuming that each sample in *f(x)* belongs to one of two models:
    - *M1* if the sample is linearly correlated to its neighbors, satisfying

$$f(x,y) = \sum_{u,v=-N}^{N} \alpha_{u,v} f(x+u, y+v)$$

    - *M2* if the sample is not correlated to its neighbors, i.e., is generated by an "outlier process".
  - Residual error $r_1(x,y) = f(x,y) - \sum_{u,v=-N}^{N} \alpha_{u,v} f(x+u, y+v)$
  - Likelihood error $w_1(x,y) = Pr(\alpha \mid r_1(x,y)) = \dfrac{e^{-r_1^2(x,y)/\sigma}}{e^{-r_1^2(x,y)/\sigma} + 1/\delta}$;

CFA Interpolation

- M – STEP
  - a new estimate of *a* is computed using weighted least squares to minimize the following quadratic error function

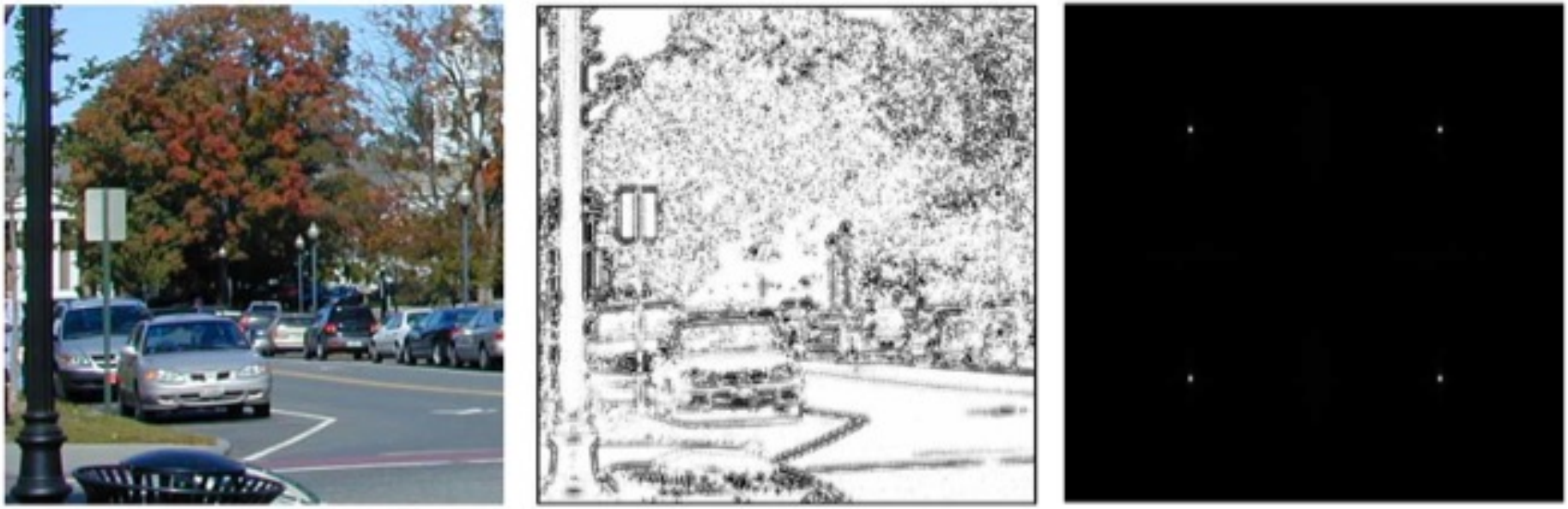
$$E(\alpha) = \sum_{x,y} w_1(x,y) \left(f(x,y) - \sum_{u,v=-N}^{N} \alpha_{u,v} f(x+u, y+v) \right)^2$$

- The E – STEP and the M – STEP are iteratively executed until a stable estimate of *a* is achieved.

CFA Interpolation

- Use the outputs of Expectation/Maximization (EM) algorithm as features to detect different interpolation.
- Two outputs:
  - The probability map. The value of each point on the probability map indicates the probability that the point is correlated with its neighbors.
  - The estimate of the weighting coefficients which represent the amount of contribution from each pixel in the interpolation kernel.
- Frequency spectrum of probability maps (peaks correspond to the periodicity in probability map which reveal the CFA correlations).
- The set of weighting coefficients obtained from an image, and the peak location and magnitudes in frequency spectrum are used as features. An SVM classifier is used.

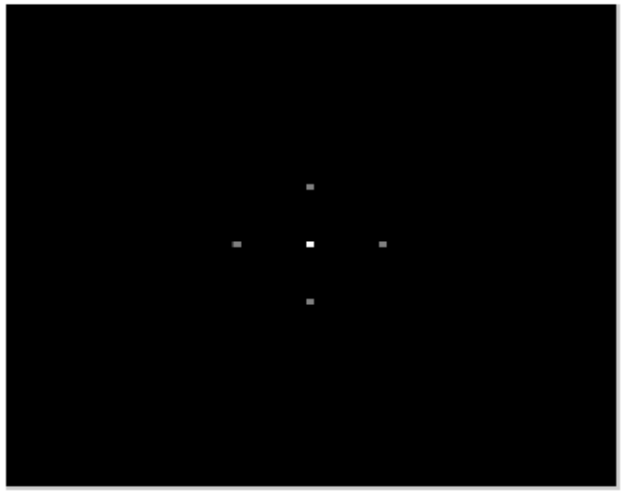
CFA Interpolation



Original CFA interpolated image and resulting CFA analysis. The peaks in the Fourier transform correspond to the periodicity in the probability map.

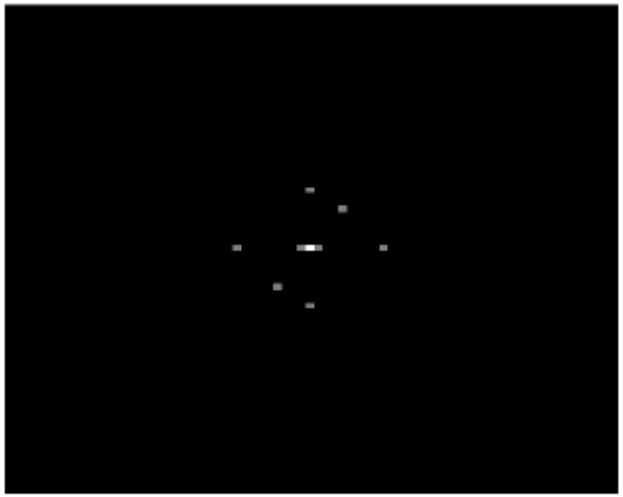
A.C. Popescu and H. Farid. Exposing digital forgeries in color filter array interpolated images. IEEE Transactions on Signal Processing, 53(10):3948-3959, 2005.

CFA Interpolation



(a) Nikon

(b) Sony



(b) Canon

Frequency spectrum of probability maps obtained by three makes of digital cameras.

Sevinc Bayram, Husrev T. Sencar, Nasir Memon, Ismail Avcibas, "Source Camera Identification Based on CFA Interpolation", Proc. of IEEE ICIP 2005.

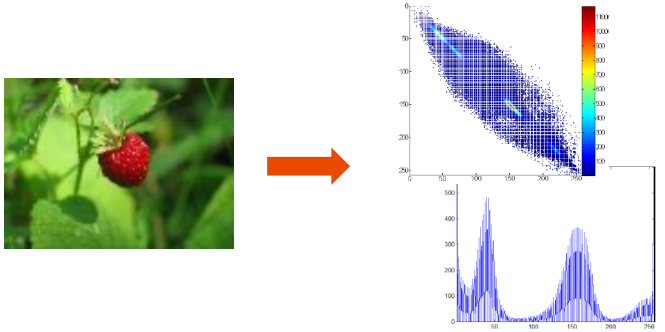
CFA Interpolation

- With this methodology it is also possible to detect and localize tampering in a portion of an image (assuming that the spliced part comes from a different device and thus it contains no or different CFA traces): windows from the CFA interpolated regions have localized peaks in the Fourier domain, whereas the windows from the "tampered" regions do not (or not the same peaks).

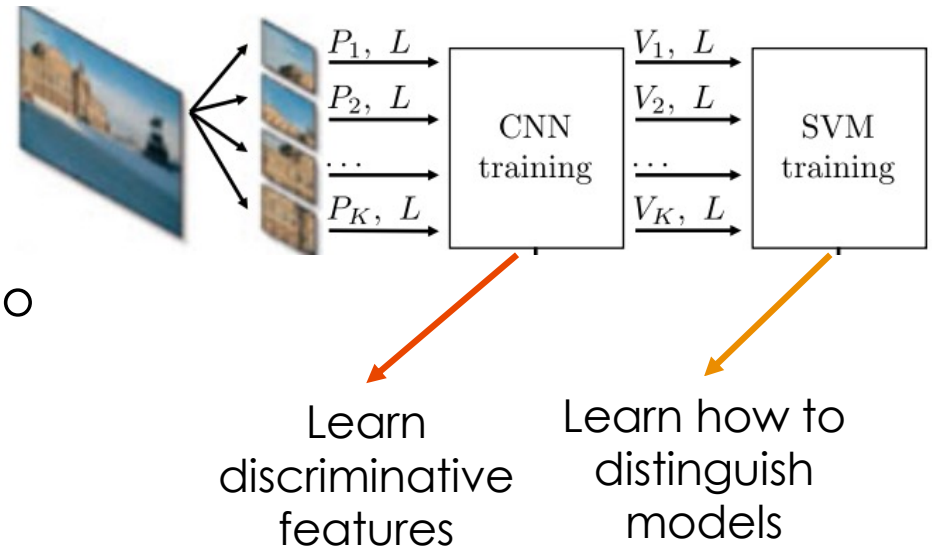
- The detection method can work also if images are subject to JPEG compression, additive noise, or luminance nonlinearities.

Signal-level Traces: deep learning

- Data Driven Approach: learn **intrinsic footprints** directly from images, rather than imposing any model or handcrafted feature recipes

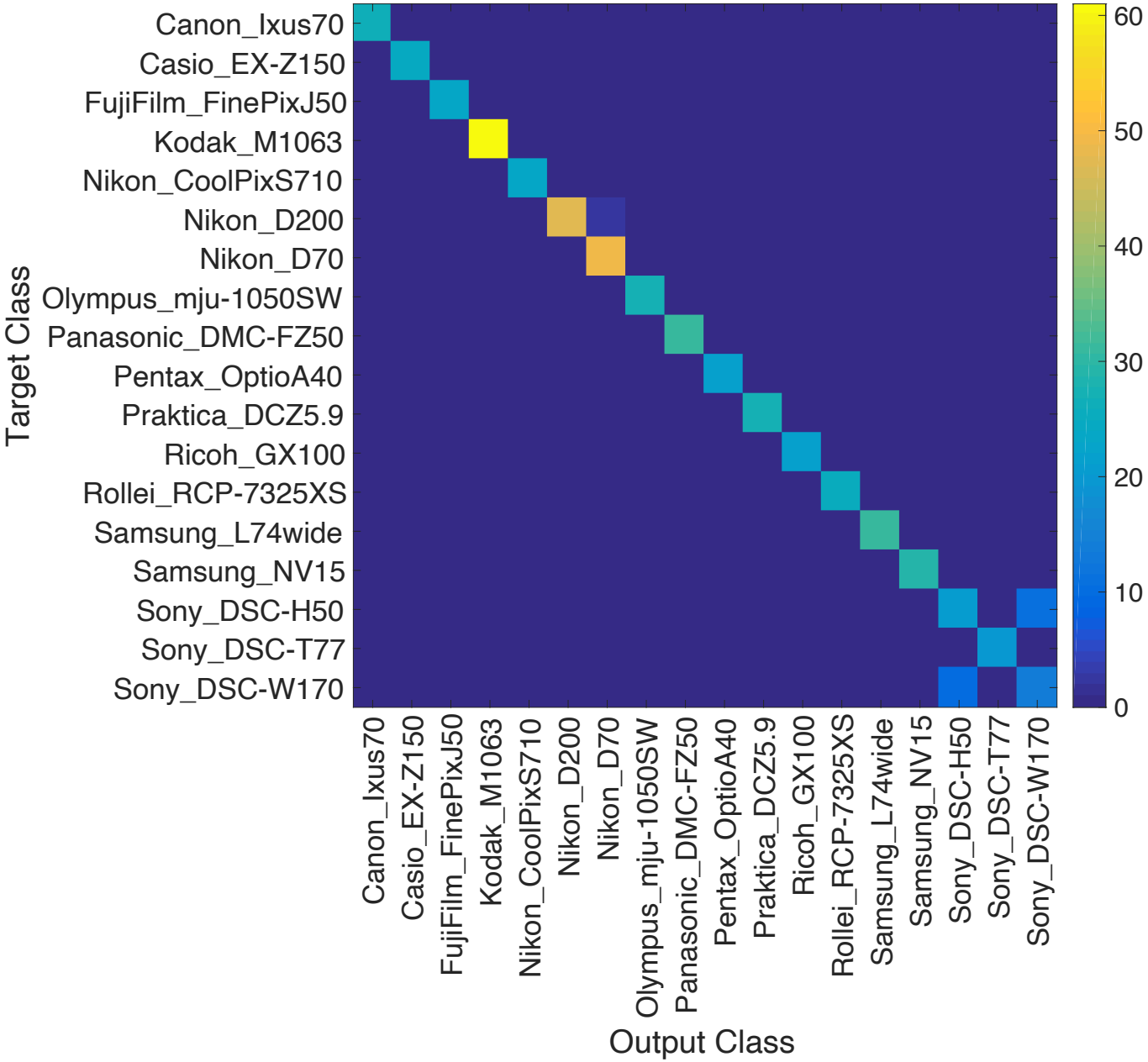


- Convolutional Neural Networks (CNNs) proved to be a good tool to learn brand/model distinctive traces



  - Learn from pixels of image patches, no need to devise complex features
  - Learned CNNs generalize well to new brand/models

Signal-level Traces: deep learning



Format-level Traces: JPEG

- Basic structures of JPEG file formats are marker segments
  - occurrence and sequence of these segments differ
- A key parameter is the quantization table
  - different brands/models usually employ different tables
- EXIF metadata including camera properties and employed image acquisition settings

marker id	short value	JIF	JFIF	EXIF	description
SOI	0xFF D8	×	×	×	start of image
APPn	0xFF En				application data
APP0	0xFF E0		×		(e.g., JFIF)
APP1	0xFF E1			×	(e.g., EXIF)
DQT	0xFF DB	×	×	×	quant. tables
DHT	0xFF C4	(×)	×	×	Huffman tables
SOF	0xFF Cn				
SOF	0xFF C0				
SOS	0xFF DA				
DRI	0xFF DD				
RSTn	0xFF Dn				
COM	0xFF FE				
EOI	0xFF D9				

$$Q = \begin{bmatrix} 16 & 11 & 10 \\ 12 & 12 & 14 \\ 14 & 13 & 16 \\ 14 & 17 & 22 \\ 18 & 22 & 37 \\ 24 & 35 & 55 \\ 49 & 64 & 78 \\ 72 & 92 & 95 \end{bmatrix}$$

Entry	Meaning
Image	
Make	Canon
Model	Canon EOS 450D
Orientation	top/left
X Resolution	72
Y Resolution	72
Resolution Unit	inch
Date Time	2011-02-27 16:14:23
YCbCr Position...	co-sited
Exif IFD Pointer	Offset: 196
Camera	
Exposure Time	1/60"
F Number	F5.6
Exposure Prog...	Portrait mode
ISO Speed Rati...	400
Exif Version	Version 2.21
Date Time Orig...	2011-02-27 16:14:23

Format level - metadata

- The JPEG compression has emerged as a near universal standard.
- JPEG is a lossy scheme, i.e. some information is lost during the process.

JPEG compression

- RGB image is transformed into luminance/chrominance space (YC_bC_r)
- Chrominance channels (C_bC_r) are subsampled by a factor of two relative to the luminance channel (Y)
- Each channel is partitioned into 8x8 pixel blocks
- These values are converted from unsigned to signed integers (e.g. from [0,255] to [-128, 127])
- Each block $f_c(\cdot)$ is converted to frequency space $F_c(\cdot)$, using a two dimensional discrete cosine transform (DCT) (c denotes a specific image channel)

$$F_c(w_k, w_l) = \sum_{m=0}^{7} \sum_{n=0}^{7} f_c(m,n) cos(w_k m) cos(w_l n) \qquad w_k = 2\pi k/8 \qquad w_l = 2\pi l/8$$

JPEG compression

- Depending on the specific frequency w_k,w_l and channel c, each DCT coefficient F_c is quantized by an amount q_c

$$\hat{F}_c(w_k, w_l) = round\left(\frac{F_c(w_k, w_l)}{q_c(w_k, w_l)}\right)$$

This stage is the primary source of data reduction and information loss (for low compression rates the values q_c tend towards 1, and increase for higher compression rates).

- Entropy encoding (typically Huffman coding): separate codes for each channel

JPEG compression

- JPEG encoders vary depending on the choice of quantization values q_c
- Quantization is specified as a 8x8 tables associated with each frequency and image channel
  - Low compression rates -> values tends toward 1
  - High compression rates -> values increase
- Typically quantization for luminance channel is less than for the other two chrominance channels;
- Lower frequencies are typically less compressed than higher frequencies.

2	1	1	1	1	1	2	1
1	1	2	2	2	2	2	4
3	2	2	2	2	5	4	4
3	4	6	5	6	6	6	5
6	6	6	7	9	8	6	7
9	7	6	6	8	11	8	9
10	10	10	10	10	6	8	11
12	11	10	12	9	10	10	10
2	2	2	2	2	2	5	3
3	5	10	7	6	7	10	10
10	10	10	10	10	10	10	10
10	10	10	10	10	10	10	10
10	10	10	10	10	10	10	10
10	10	10	10	10	10	10	10
10	10	10	10	10	10	10	10
10	10	10	10	10	10	10	10
2	2	2	2	2	2	5	3
3	5	10	7	6	7	10	10
10	10	10	10	10	10	10	10
10	10	10	10	10	10	10	10
10	10	10	10	10	10	10	10
10	10	10	10	10	10	10	10
10	10	10	10	10	10	10	10
10	10	10	10	10	10	10	10

A sample JPEG quantization table employed by a Nikon Coolpix 2500 digital camera

JPEG compression

- Specific quantization tables and Huffman codes are embedded into the JPEG header.
- JPEG quantization tables and Huffman codes, along with other data extracted from the header are a distinct camera signature which can be used for authentication.
- Indeed, the JPEG standard does not enforce any specific quantization table or Huffman code. Camera and software are free to balance compression and quality to their own needs and tastes.

JPEG Header

- Full resolution image - 284 extracted values
  - 2 image dimensions: used to distinguish between cameras with different sensor resolutions
  - 192 quantization values (8 x 8 quantization tables for each of the three channels)
  - 90 Huffman codes (6 sets of 15 values corresponding to the number of codes of length 1,2,...15. Each channel needs two codes, one for the DC coefficient and one for AC coefficients)

JPEG Header

- Thumbnail image
  - Thumbnail version of the full resolution image is embedded into the JPEG header (cropping, filtering and down-sampling)
  - Not all the cameras create a thumbnail, thus characterizing the camera.
- 284 extracted values:
  - 2 thumbnail dimensions
  - 192 quantization values (8 x 8 quantization tables for each of the three channels)
  - 90 Huffman codes (6 sets of 15 values corresponding to the number of codes of length 1,2,...15. Each channel needs two codes, one for the DC coefficient and one for AC coefficients)

JPEG Header

- EXIF header
  - it stores information about the camera and image
  - 5 main image files directories (IFDs) (Primary, Exif, Interoperablity, Thumbnail, GPS)
  - Additional IFDs
  - Camera manufacturer customize their metadata, leading to parsing errors, which are considered a feature of the camera design.
- 8 extracted values
  - 5 entry counts from the standard IFDs
  - 1 for additional IFD
  - 1 for the number of entries in these additional IFDs
  - 1 for the number of parser errors

JPEG Header

- 284 from the full resolution image
- 284 from the thumbnail
- 8 from the EXIF
- TOTAL: 576 values used for authentication!
- Basic idea: any photo-editing software will alter this original signature and can therefore be detected ➠ Extract the signature from an image and compare it to a database of known authentic camera signatures
  - any matching camera make and model can be compared to the make and model specified in the image's EXIF metadata.
  - any mismatch is strong evidence of some form of tampering.

	Canon EOS Rebel XTi	Nikon D40
image dimensions	2592 × 3888	2000 × 3008
image quantization table (Y)	1 1 1 1 1 2 3 3 1 1 1 1 1 3 3 3 1 1 1 1 2 3 3 3 1 1 1 1 3 4 4 3 1 1 2 3 3 5 5 4 1 2 3 3 4 5 6 5 2 3 4 4 5 6 6 5 4 5 5 5 6 5 5 5	1 1 1 1 1 1 1 2 1 1 1 1 1 2 2 2 1 1 1 1 1 2 2 2 1 1 1 1 1 2 2 2 1 1 1 2 2 3 3 2 1 1 2 2 2 3 3 3 1 2 2 2 3 3 3 3 2 3 3 3 3 3 3 3
image quantization table (Cb)	1 1 1 2 5 5 5 5 1 1 1 3 5 5 5 5 1 1 3 5 5 5 5 5 2 3 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5	1 1 1 1 3 3 3 3 1 1 1 2 3 3 3 3 1 1 2 3 3 3 3 3 1 2 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
image quantization table (Cr)	1 1 1 2 5 5 5 5 1 1 1 3 5 5 5 5 1 1 3 5 5 5 5 5 2 3 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5	1 1 1 1 3 3 3 3 1 1 1 2 3 3 3 3 1 1 2 3 3 3 3 3 1 2 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
image Huffman code DC (Y)	1 5 1 1 1 1 1 1 0 0 0 0 0 0 0 0	1 5 1 1 1 1 1 1 0 0 0 0 0 0 0 0
image Huffman code DC (Cb)	3 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0	3 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0
image Huffman code DC (Cr)	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
image Huffman code AC (Y)	2 1 3 3 2 4 3 5 5 4 4 0 0 1 125	2 1 3 3 2 4 3 5 5 4 4 0 0 1 125
image Huffman code AC (Cb)	2 1 2 4 4 3 4 7 5 4 4 0 1 2 119	2 1 2 4 4 3 4 7 5 4 4 0 1 2 119
image Huffman code AC (Cr)	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
thumbnail dimensions	120 × 160	120 × 160
thumbnail quantization table (Y)	3 2 2 3 5 8 10 12 2 2 3 4 5 11 11 13 3 2 3 5 8 11 13 11 3 3 4 6 10 17 15 12 3 4 7 11 13 21 20 15 5 7 10 12 15 20 21 17 9 12 15 17 20 23 23 19 14 17 18 19 21 19 20 19	1 1 1 1 1 2 3 4 1 1 1 1 2 3 4 3 1 1 1 1 2 3 4 3 1 1 1 2 3 5 5 4 1 1 2 3 4 6 6 5 1 2 3 4 5 6 7 5 3 4 5 5 6 7 7 6 4 5 6 6 7 6 6 6
thumbnail quantization table (Cb)	3 3 5 9 19 19 19 19 3 4 5 13 19 19 19 19 5 5 11 19 19 19 19 19 9 13 19 19 19 19 19 19 19 19 19 19 19 19 19 19 19 19 19 19 19 19 19 19 19 19 19 19 19 19 19 19 19 19 19 19 19 19 19 19	1 1 1 3 6 6 6 6 1 1 2 4 6 6 6 6 1 2 3 6 6 6 6 6 3 4 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6
thumbnail quantization table (Cr)	3 3 5 9 19 19 19 19 3 4 5 13 19 19 19 19 5 5 11 19 19 19 19 19 9 13 19 19 19 19 19 19 19 19 19 19 19 19 19 19 19 19 19 19 19 19 19 19 19 19 19 19 19 19 19 19 19 19 19 19 19 19 19 19	1 1 1 3 6 6 6 6 1 1 2 4 6 6 6 6 1 2 3 6 6 6 6 6 3 4 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6
thumbnail Huffman code DC (Y)	1 5 1 1 1 1 1 1 0 0 0 0 0 0 0 0	1 5 1 1 1 1 1 1 0 0 0 0 0 0 0 0
thumbnail Huffman code DC (Cb)	3 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0	3 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0
thumbnail Huffman code DC (Cr)	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
thumbnail Huffman code AC (Y)	2 1 3 3 2 4 3 5 5 4 4 0 0 1 125	2 1 3 3 2 4 3 5 5 4 4 0 0 1 125
thumbnail Huffman code AC (Cb)	2 1 2 4 4 3 4 7 5 4 4 0 1 2 119	2 1 2 4 4 3 4 7 5 4 4 0 1 2 119
thumbnail Huffman code AC (Cr)	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
EXIF count	9 28 2 6 0 7 162 0	10 40 2 7 0 7 94 0

M.K. Johnson, E. Kee and H. Farid. Digital image authentication from JPEG headers. IEEE Transactions on Information Forensics & Security, 2011

JPEG Header

- 1.3 million images - 9163 different pairings of camera make, model and signature, representing 33 camera manufacturers and 773 different cell-phones and camera models.
- Equivalence class -> camera sharing the same configuration.
- Parameters not highly correlated; hence their combination improves overall distinctiveness.
- Photoshop signature is unique.

	Equivalence Class Size						
	1	2	3	4	5	median	max
image	12.9%	7.9%	6.2%	6.6%	3.4%	11	185
thumb	1.1%	1.1%	1.0%	1.1%	0.7%	694	960
EXIF	8.8%	5.4%	4.2%	3.2%	2.6%	25	188
image+thumb	24.9%	15.3%	11.3%	7.9%	3.7%	3	91
all	69.1%	12.8%	5.7%	4.0%	2.9%	1	14

Open issues

- Most of solutions are tested on Perfect knowledge scenarios only
  - What if the content comes from **unknown devices**?
- How to deal with a **new device**?
  - Classifier needs re-training for adding new models
- Camera identification for **videos** is still unexplored
  - …while they are becoming the most diffused information