Sevinc Bayram, Husrev T. Sencar, Nasir Memon

Classification of digital camera-models based on demosaicing artifacts

Presented by Jevin Sweval

Background

- Computer forensics has several concerns when dealing with images
  - Is the image real? (i.e. is the image computer generated)
  - Is the image doctored?
  - **What can we say about the image acquisition device?**

Background

- In computer forensics, it is often handy to match data with the device that produced it, in the same vein as ballistic fingerprinting
  - Discover the printer that printed some leaked documents
  - Reveal the microphone/recorder combo used to record a revealing admission
  - **Uncover the camera used to take an espionage surveillance photo**

Background

- Metadata is often helpful in tracking down the origins of a document
  - Have you ever looked at the metadata of a Microsoft Word document? (contains name, initials, organization)
  - JPEG EXIF data often contains time, location, camera, lens settings
  - **Careful individuals will scrub or fake sensitive metadata**

Acquisition device recognition

- If the image metadata is unavailable, how do you determine the source of an image?
- You must rely upon characteristics of the generated image to make informed guesses about its source

Acquisition device characteristics

- Characteristics unique to a camera model
  - Camera components
    - Lens, imaging sensor, optical filters
    - Different brands often buy the same components!
  - Imaging pipeline
    - Imaging ASIC, imaging algorithms (**demosaicing**, resizing, color calibration), compression algorithms
    - Found to be more unique than camera components
- Characteristics unique to an individual device
  - Image sensor noise
  - Dead/stuck pixels
  - Dust specs trapped between lens and sensor

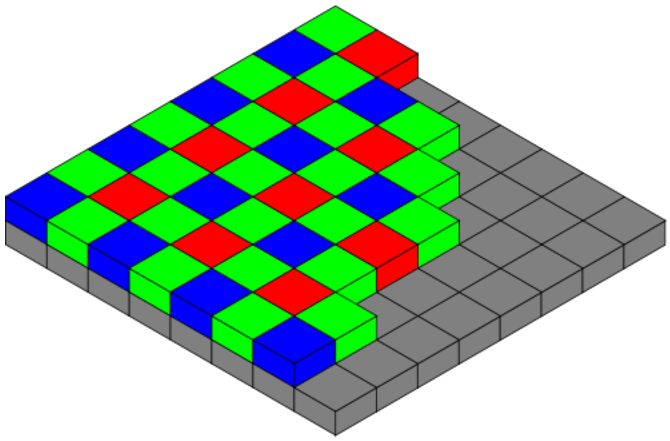
Acquisition device characteristics

- Characteristics unique to a camera model
  - Often these characteristics are found in *other* camera models due to similar components
- Characteristics unique to an individual device
  - These characteristics are often minute
  - May change over time and with different operating conditions

Your camera lied to you!

- You thought your camera had X pixels? You are only being told half the story! (w.r.t. consumer cameras)
- While your camera really does have X pixels as stated, each of those pixels only "sees" one color (usually RGB, sometimes CYGM). You don't get X full-color pixels but rather X full-color pixels *after software interpolation*.

Color filter arrays and demosaicing



Bayer filter

Color filter arrays and demosaicing

- Bayer filter
  - By far the most common filter in consumer cameras
  - Twice as many green as red or blue pixels
    - Humans are most sensitive to green light

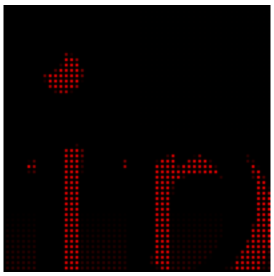
Color filter arrays and demosaicing

Real Image

Color filter arrays and demosaicing

Bayer Pattern
(raw data from sensor)

Color filter arrays and demosaicing



R

G

B

Color filter arrays and demosaicing

Real Image

Demosaiced (interpolated) Image

Note some loss of sharpness and color fidelity

Color filter arrays and demosaicing

- This paper focuses on identifying, detecting, and classifying traces of demosaicing
- Some similar previous research has tried characterizing the up-sampling algorithm, the second-derivative of demosaicing, or fingerprinting persistent image sensor noise

Demosaicing

- Demosaicing is really just interpolation based on color
- Simple resizing interpolations are suitably adapted for demosaicing purposes in smooth areas of the image
- Demosaicing algorithms are more complicated in “busy” areas of the image in order to reduce artifacts
  - This paper’s methods focused on analyzing smooth areas of the image because the interpolation fingerprints were more easily observed in those areas

Simple Demosaicing

- These methods use popular interpolation techniques like nearest-neighbor or bilinear interpolation
- All color channels are treated as completely separate images and are "blended" together
- As stated, this works well for smooth areas of the image but produce unsuitable artifacts for more complicated portions of images

Advanced Demosaicing

- The simple demosaicing methods fail to utilize the fact that the different color channels are highly correlated with each other
- These techniques utilize edge-detection, hue-hints and anti-aliasing techniques

Characterizing demosaicing artifacts

- Two methods were used to detect and classify demosaicing artifacts
  - Images were run through an Expectation-Maximization algorithm to determine the original interpolation kernel
  - Second-derivatives were taken of the images and used to determine the interpolation factors

Expectation-Maximization

- An iterative algorithm that flip-flops between two steps (expectation and maximization) until convergence
- Lets us converge on a probable interpolation kernel given the observed color values

Expectation-Maximization

- Expectation step
  - Generate an expectation (x) using current estimates of parameters (p) conditioned upon the observations (y)
  - $x_{t+1} = E[x_t|y, p_t]$

Expectation-Maximization

- Maximization step
  - Using the expectation (x) as if it were true values, determine the parameters (p) that are most likely to create x
  - $p_{t+1} = F[x_t, y]$
    - F here is a function that gives the the maximum likelihood of the parameters p given the expected values x and observed values y

Expectation-Maximization

- Interpolation kernel β is desired
  - Must chose a kernel size before performing EM
  - Sizes 3x3, 4x4, 5x5 tried, 5x5 was most accurate
  - Kernel is used as a feature to detect the camera model
- This paper ran the EM algorithm on just the red channel because it is more heavily interpolated than the green channel
  - More interpolation => stronger artifacts
  - Blue channel would work equally well

Second-order derivative

- Interpolation introduces periodicity in the variance of the second-order derivative of an image
- This periodicity is equal to the interpolation factor and can be used as another feature to discriminate camera models

Second-order derivative

- Second-order derivative is calculated on a row-by-row basis
  - $sd_r(i,j) = 2r(i,j) – r(i,j+1) + r(i,j-1)$
  - $r(i,j)$ are the red color values for pixel (i,j)
- The magnitude of each row is taken
- The pseudo-variance of the row-magnitudes is calculated
  - $v(j) = \text{sum}(|sd_r(i,j)|, i = 0, i < \text{height})$

Second-order derivative

- JPEG compression uses an 8x8 DCT
  - Ignore variance periods of 8
- The interpolation rate is used as a feature for discrimination
- The magnitudes of the variance peaks are another utilized feature
- The correlation between variance peaks of different color channels is the final feature used for discrimination

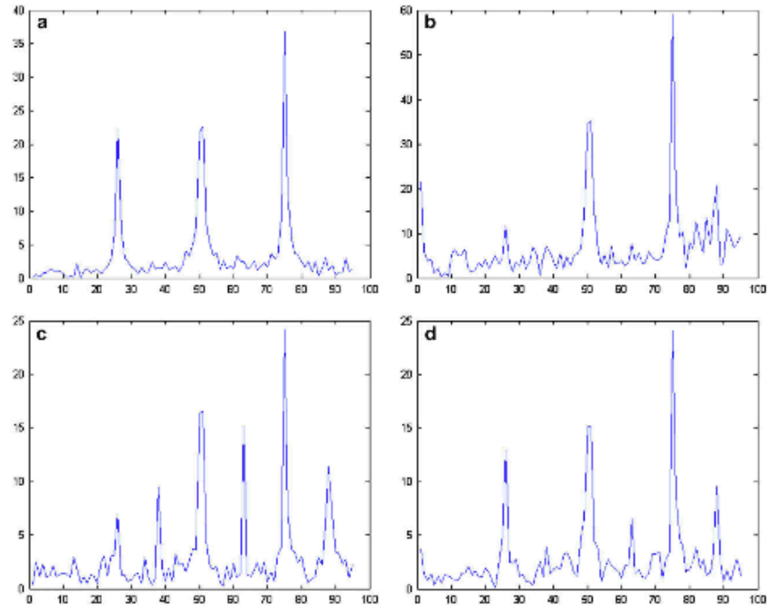


Fig. 3 – Frequency spectrum of averaged second-order derivatives corresponding to JPEG compression and the three models of digital cameras with JPEG output images. (a) Peaks due to JPEG compression (b) Canon Powershot S200 (c) Sony DSC-51 (d) Nikon E-2100.

Noise analysis

- Every image sensor has some inherit noise pattern
  - Can be seen by taking a long exposure with the lens cap on
  - May vary with temperature, long spans of time
    - For each camera, many images were averaged together to determine a noise profile

Noise analysis

- Previous work by the authors correlated the noise profile from a given image to measured noise profiles of many cameras in order to determine the source device
  - Some cameras had very high false-positive rates
- Noise profile matching can be combined with demosaicing matching to improve false-positive rate

Noise analysis

- First use noise profile matching to determine if the image is from a given source
- If the noise profile matching is affirmative, further check with demosaicing matching

Experimental Results

- SVM classifier was used to classify an image given a feature set
- Full feature set consisted of interpolation kernel coefficients (determined via EM) and second-order differentiation features (interpolation rate, variance peak magnitudes, cross-channel correlation of variance peaks)

Experimental Results

- Sequential forward floating search (SFFS) was used to determine a subset of the full feature set for analysis
- SFFS works by adding the most significant features then pruning features until performance (determined using SVM) degrades

Experimental Results – Part 1

- Five camera models
- Same scene photographed 200 times for each camera
  - Using the same scene prevents textural attributes of the scene from disturbing the results
  - 100 images used to train the SVM, 100 used for testing the classifier

Experimental Results – Part 1

- When discriminating between 4 cameras, detection accuracy was 88%
- When discriminating between all 5 cameras, detection accuracy was 84.8%

Predicted Model

Actual Model

	Canon	Datron	HP	Sony
Canon	84	7	2	6
Datron	4	87	1	3
HP	9	6	93	3
Sony	3	0	4	88

Experimental Results – Part 1

- With all five cameras, the Canon and Kodak models were often confused with each other
  - These models likely have similar demosaicing algorithms

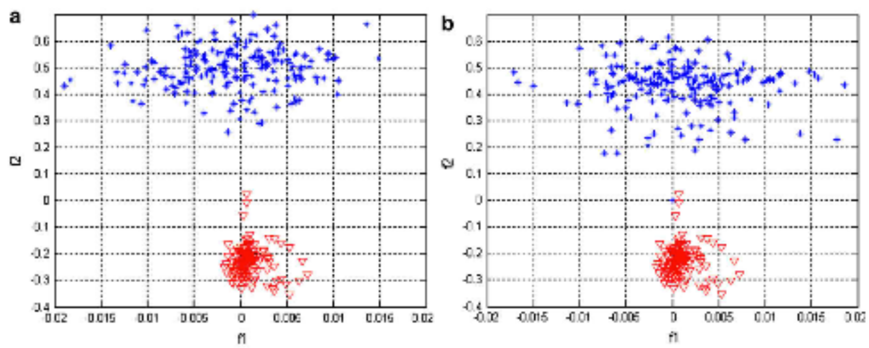
Predicted Model

Actual Model	Canon	Datron	HP	Kodak	Sony
Canon	73	6	0	10	8
Datron	4	88	0	3	1
HP	0	0	96	4	1
Kodak	16	4	2	78	5
Sony	7	2	2	5	85

Experimental Results – Part 2

- 600 pictures were taken from each camera depicting various scenes
  - 300 used for SVM training, 300 for analysis
- When determining if a picture was taken by one of two cameras, the prediction accuracy was 100%
- When identifying the source among 3 cameras, accuracy was 92.6%

Experimental Results – Part 2



Good feature clustering
between two camera models

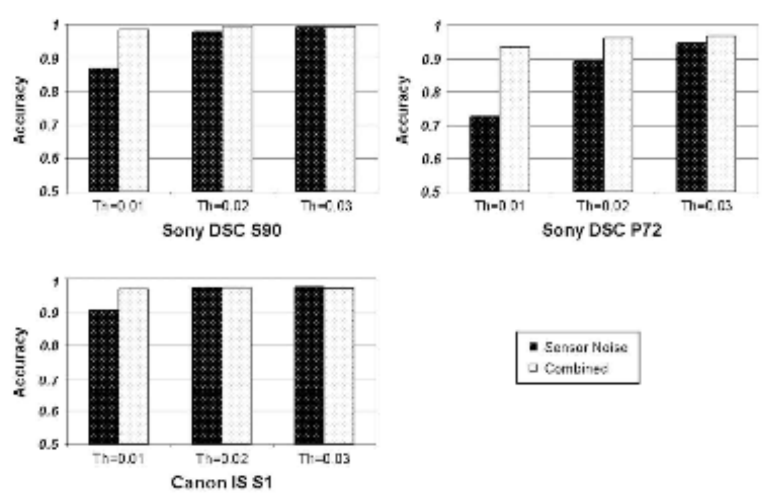
Experimental Results – Part 3

- The final experiment tries to determine whether a particular camera is the source device for a given image using both noise pattern matching and demosaicing artifact matching
- 300 images used for determining noise patterns and demosaicing classification training
- 300 more images from the given camera along with 5000 images from other cameras were used for testing

Experimental Results – Part 3

- The following graph shows the experimental results (prediction that the camera took a given photo) for various noise pattern matching thresholds
  - Lower threshold => more imprecise match
- Combining noise matching with demosaicing matching greatly improved false-positive rates with little decrease in true-positives

Experimental Results – Part 3



Conclusions

- Demosaicing classification can predict the source device of an image with ~ 90% accuracy
- Demosaicing classification increase accuracy of source device identification using noise pattern matching by ~6%
- Open question: how to make these methods more robust against malicious-processing