Aalto University
School of Science
Degree Programme in Computer Science and Engineering

Henrik J. Asplund

Imaging sensor identification with photo response non-uniformity fingerprints

Master's thesis
Espoo, November 26, 2015

Supervisor: Professor Eljas Soisalon-Soininen, Aalto University
Advisor: Mervi Ranta, M.Sc., Idegos Ltd.

Aalto University
School of Science
Degree Programme in Computer Science and Engineering

ABSTRACT OF
MASTER'S THESIS

Author:	Henrik J. Asplund		
Title:	Imaging sensor identification with photo response non-uniformity fingerprints		
Date:	November 26, 2015	**Pages:**	82+10
Major:	Software Engineering	**Code:**	T-109
Instructor:	M.Sc. Mervi Ranta		
Supervisor:	Professor Eljas Soisalon-Soininen		

This thesis shows a method to identify a camera source by examining the noise inherent to the imaging process of the camera. The noise is caused by the imaging hardware, e.g. physical properties of charge-coupled device (CCD), the lens, and the Bayer pattern filter. The noise is then altered by the algorithms of the imaging pipeline. After the imaging pipeline, the noise can be isolated from the image by calculating the difference between noisy and denoised image.

Noise can be used to form a camera fingerprint by calculating mean noise of a number of training images from same camera, pixel by pixel. The fingerprint can be used to identify the camera by calculating the correlation coefficient between the fingerprints from the cameras and a test image. The image is then assigned to the camera with highest correlation.

The key factors affecting the recognition accuracy and stability are the denoising algorithm and number of training images. It was shown that the best results are achieved with 60 training images and wavelet filter.

This thesis evaluates the identifcation process in four cases. Firstly, between cameras chosen so that each is from different model. Secondly, between different individual cameras from the same model. Thirdly, between all individual cameras without considering the camera model. Finally, forming a fingerprint from one camera from each model, and then using them to identify the rest of the cameras from that model.

It was shown that in the first two cases the identification process is feasible, accurate and reasonably stabile. In the latter two cases, the identification process failed to achieve sufficient accuracy to be feasible.

Keywords: camera identification, source identification, sensor identification, image filtering, image classification, noise patterns, machine learning, sensor fingerprints, experimentations, methodological validation, classification problems

Language: English

Aalto-yliopisto
Perustieteiden korkeakoulu
Tietotekniikan koulutusohjelma

DIPLOMITYÖN
TIIVISTELMÄ

Tekijä:	Henrik J. Asplund		
Työn nimi:	Kuva-anturien tunnistaminen valovasteen epäyhdenmukaisuutta hyödyntäen		
Päiväys:	26.11.2015	**Sivuja:**	82+10
Pääaine:	Ohjelmistotekniikka	**Koodi:**	T-109
Ohjaaja:	DI Mervi Ranta		
Valvoja:	Professori Eljas Soisalon-Soininen		

Tässä työssä esitetään menetelmä kuvalähteenä olevan kameran tunnistamiseksi tutkimalla kuvausprosessissa sinällään syntyvää kohinaa. Kohina syntyy kuvauksessa käytettävästä laitteistosta, esim. kuva-anturista (CCD), linssistä ja Bayer-suotimesta. Kohinaa muokkaavat kameran automaattisesti kuvanparannukseen käyttämät algoritmit. Kuvanparannuksen jälkeen kohinan voi eristää muodostamalla erotuksen kohinan sisältävän kuvan ja suodatetun kuvan välillä.

Kameran sormenjäljen voi muodostaa laskemalla pikseleittäin keskiarvon opetuskuvien kohinasta. Sormenjälkeä käytetään laskemaan korrelaatio testikuvan ja sormenjäljen välillä. Kuvan ottaneeksi kameraksi tunnistetaan se, jonka sormenjäljen ja testikuvan kohinan välillä on suurin korrelaatio. Tärkeimmät tunnistuksen tarkkuuteen ja vakauteen vaikuttavat tekijät ovat kohinanpoistoalgoritmi ja opetuskuvien määrä. Työssä osoitetaan, että parhaat tulokset saadaan käyttämällä 60:tä opetuskuvaa ja aallokesuodatusta.

Tässä työssä arvioidaan tunnistusprosessia neljässä tapauksessa. Ensiksi eri malleista valittujen yksittäisten kameroiden suhteen, toiseksi saman kameramallin yksilöiden välillä, kolmanneksi kaikkien yksittäisten kameroiden välillä jättäen huomiotta kameramallin, ja viimeiseksi pyritään yhtä kameraa käyttäen muodostamaan prototyyppisormenjälki, jolla tunnistaa muut samanmalliset kamerat.

Työssä osoitettiin, että kahdessa ensinmainitussa tapauksessa tunnistus toimii riittävän tarkasti ja vakaasti. Jälkimmäisessä kahdessa tapauksessa tunnistus ei saavuttanut riittävää tarkkuutta.

Avainsanat: kamerantunnistus, lähteen tunnistus, anturintunnistus, kuvansuodatus, kuvien luokittelu, kohinakuviot, koneoppiminen, anturien sormenjäljet, koejärjestelyt, menetelmällinen todistaminen, luokitteluongelmat

Kieli: englanti

Contents

List of Figures

List of Tables

Acronyms

CCD charge coupled devices

CFA color filter array

CMYK cyan, magenta, yellow and black

DWT discrete wavelet transform

ELM Extreme Learning Machine

EXIF Exchangeable Image File Format

FFT Fast Fourier Transform

FPN fixed pattern noise

FT Fourier Transform

IDWT inverse discrete wavelet transform

IFFT Inverse Fast Fourier Transform

IFT Inverse Fourier Transform

JPEG Joint Photographic Experts Group

LFD low frequency defects

LTI linear time-invariant

PDE partial differential equation

PNU pixel non-uniformity noise

PRNU photo-response non-uniformity

RGB red, green and blue

TIFF Tagged Image FIle Format

Mathematical notation

$F(I_k)$	k^{th} image filtered with filter F
$\mathcal{F}\{x(t)\}$	Fourier transform of function x
$\mathcal{F}^{-1}\{X(i\omega\, t)\}$	Inverse fourier transform of function X
I_k	k^{th} image
$\mathbf{I}$	unity matrix, not to be confused with I_k
ℓ	filter mask side length
μ	statistical mean
N_{ref}	Number of reference images
N_{test}	Number of test images
$\rho^C(I_k)$	correlation between the reference pattern of camera C and image I_k
$P(\omega_i\|\mathbf{x_i})$	posterior probability for $\mathbf{x_i} \in \omega_i$, i.e. probability of class ω_i given observation $\mathbf{x_i}$
$P(\omega_i)$	a priori probability for class ω_i
$P(\mathbf{x_i})$	probability of observation $\mathbf{x_i}$
ϕ	orthogonal wavelet scaling function
ψ	orthogonal wavelet mother wavelet function
ϕ_d	biorthogonal wavelet decomposition scaling function
ψ_d	biorthogonal wavelet decomposition mother wavelet function
ϕ_r	biorthogonal wavelet reconstruction scaling function
ψ_r	biorthogonal wavelet reconstruction mother wavelet function
σ	statistical standard deviation
$\mathbf{W_k}$	difference of unfiltered and filtered k^{th} image
$\bar{\mathbf{W}}_\mathbf{k}$	the mean of $\mathbf{W}_k$
$\mathbf{W^C_{ref}}$	the reference pattern of camera C
$\bar{\mathbf{W}}^\mathbf{C}_\mathbf{ref}$	the mean of reference pattern $\bar{\mathbf{W}}^C_{ref}$

Chapter 1

Introduction

Amount of information available has been rapidly growing for years, and more and more of it is stored in form of videos, images and audio. Recognizing the contents and subjects of the images is a wide and successful area of technology, beginning from photo albums and databases, and now moving towards face recognizing security systems and presenting augmenting information.

Another direction for utilizing the image information is to disregard the contents and subjects altogether, and concentrate on the sensors and devices producing it. In this direction an image is just a bunch of data that contains hidden information on the sensor that produced it. The origin of image is an answer to question "which sensor produced this image and how?", as opposed to the first research direction's question "who took this image and who are in it?". This is mostly uncharted territory, and still a huge effort is needed to determine what can be said of the originating sensor based on just an array of pixels and nothing more. It is possible to use images to derive information that was not intentionally put there by the person who took the image, and information that the person is not aware of being there.

Interesting in this second approach is that current research is mostly based on exploiting the imperfections of the sensors. In the first approach, the imperfections are unwanted and sometimes even have a deeply deteriorating effect on the results. However, as the subject matter of this work is sensor recognition, the imperfections are a necessity. No imaging device is perfect – the minor aberrations and flaws in lenses, the unwanted currents traveling through the imaging sensor, the algorithms of the manufacturers generating artifacts, are just a few examples of the imperfections. These imperfections are partly unique for the individual imaging device, and partly affect all the cameras from a manufacturer, or all the cameras of the same model. To summarize, no imaging device is perfect, and no two imaging devices have exactly the same imperfections. This thesis shows how one type of the imperfections, pattern non-uniformity, can be exploited to identify the original imaging device.

Promising tools and methods are based largely on machine learning. Even some of the simple methods, e.g., naïve Bayesian classification or linear regression, can produce rather accurate results. More advanced methods, such as neural networks are also viable, but with additional requirements for computing power. Choosing the method is not straightforward, as can be expected. While the simpler methods are faster, they require a larger amount of training images. Neural networks can do with fewer training images, but is slower. This tradeoff has a considerable effect on choosing the methods, since processing and filtering high resolution images is not a trivial problem even nowadays - it requires time and patience.

The key for successful image classification, in addition to the classification method itself, is the preprocessing of the images. Again, there is a tradeoff. While more advanced filtering techniques provide better results in eliminating the image content from the noise patterns, they also require time. Vice versa, simpler filtering methods provide poor quality noise patterns, but are faster to run. Perhaps the most important factor for choosing the filtering technique is whether the filtering has to be done while classification, e.g., in the case of having to constantly add new images, or whether the images can be processed before classification. If the images can be processed before classification, higher grade filtering is suitable. If not, some kind of balance has to be found to provide sufficient quality of noise patterns while keeping the required time for filtering small enough. Again, choosing the filtering is not a trivial matter.

As can be seen, one of the hardest problems in choosing the filtering and classification methods is finding a balance so that noise patterns have good enough quality, and the classification method can achieve high enough accuracy with the patterns. There are additional factors for choosing the methods, e.g., image compression, size, quality, storage — network or hard disk, and properties of the source camera and lenses. Understanding the camera imaging pipeline is the key to analyze these factors. Finally, the experimentations in this thesis can be used as a template in order to find a viable combination of methods.

This work uses cameras as an example, but the principle can be applied to e.g. scanners. In order to understand the sources of imperfections it is necessary to familiarize oneself with the camera imaging pipeline. The imperfections produce noise in the image data, and the specific patterns of noise form fingerprints that can be used to track image origin. However, in order to start developing applications, it is first necessary to carry out experimentations in order to validate the idea – specifically, that cameras leave fingerprints to images, and the fingerprints can be detected and exploited in recognizing the originating sensor.

Potential applications for the sensor source recognition are in the field of forensics and digital rights management. However, the usage of a recognition system in these fields have serious implications – unvalidated techniques and unreliable results can lead to wrongful convictions and lawsuits. Since product development in these fields require reliability, it is of utmost importance to properly validate the methods and results. The consequences "proof of concept" and "seems to work" approaches can lead to disaster.

This thesis does not consider application or product development based on the experimentations. The idea is to give a template of experimentations required before product development can be commenced. Thus, the results of this thesis validate the give combinations of methods, but does not argue for or against using them in a commercial product. In product development, also other factors than purely methodological feasibility have to be considered, such as computing resources, required memory, image storage, utilization of network and so on. However, the experimentations presented in this thesis are used to validate that the crucial features, the core of the classification process, is feasible.

First, chapter 2 presents the digital imaging pipeline. Chapter 3 presents the classification and structure of sensor pattern noise. Chapter 4 gives an introduction to the image filtering methods used in this thesis.Chapter 5 presents how the noise patterns are calculated in practice, and how the fingerprints are generated, how the correlations between fingerprints and noise patterns are calculated, and how they can be processed for classification. Chapter 6 introduces three solutions to the classification, i.e., source recognition problem. Chapter 7 presents the focus and the hypotheses of the experimentation. Chapter 8 shows how the data is collected for the experimentations, and the semantics of the images and cameras in the classification process - tiered, flat or pooled. The results of the experimentations are presented in chapter 9. Finally, chapter 10 presents the conclusions and and further work.

Chapter 2

Digital imaging pipeline

The methods for source camera identification utilize the knowledge on how the different phases of digital image acquisition pipeline are performed. Therefore, the key to understanding them is comprehension of digital image acquisition pipeline. Each phase of the pipeline generates artifacts that can be detected, measured and exploited in order to recognize the source camera, or at least the camera model.

Figure 2.1 shows the phases of image acquisition pipeline. The phases are:

1. *Sensor, aperture and lens*: physical measurement of light and focusing the image
2. *Pre-processing*: correcting the image by removing typical problems and making the image visually pleasing
3. *White balance*: correcting the colors of the measured image signal
4. *Demosaicking*: consolidating the pixel colors and removing artifacts from color filter array (CFA)
5. *Color transformations*: rendering the image to a color space that allow for techniques used in postprocessing stage
6. *Post-processing*: further enhancements to the image, e.g. removing superfluous pixel information and artifacts from previous phases
7. *Compress and store*: rendering the image to the desired color space, and compressing and storing image in desired format in a permanent storage medium.

The imaging pipeline may seem rather complex due to the several phases in which the image is processed and undesirable effects are removed. The main reason for the complexity is that the corrections in previous phases cause undesirable artifacts themselves. Those artifacts, however, can be controlled and the image signal containing them is easier to process than the original.

Sensor, aperture and lens

Since there are three different color bands that need to be measured, i.e., red, green and blue, each pixel would need three different kinds of sensors for the respective

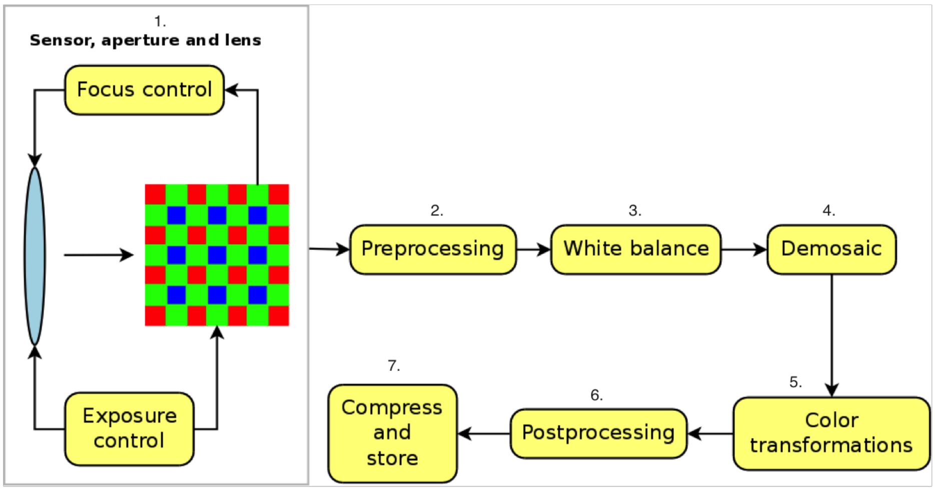
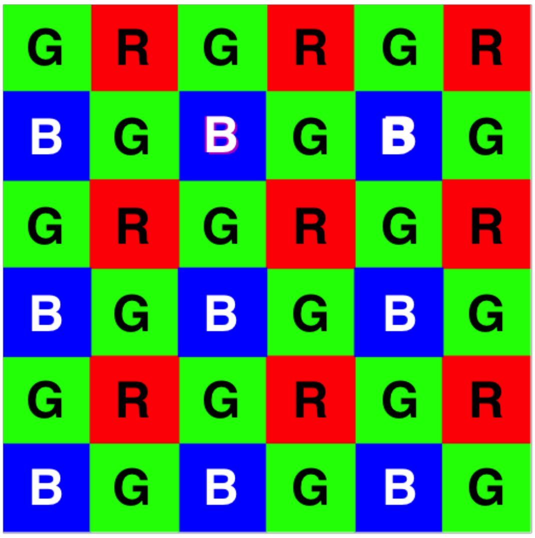


Figure 2.1: Digital camera imaging pipeline (Ramanath et al., 2005)

wavelengths. Production of this kind of an imaging sensor array is not cost effective. A solution is using a color filter array (CFA) to filter the wavelengths so that the underlying pixel sensors need to measure just one wavelength. The most common CFA pattern is the Bayer pattern that can be seen in Figure 2.2. In the figure, colors are annotated as G=green, R=red and B=blue. As can be seen in the figure, half of the CFA is covered with green filters, and the rest of the array is equally divided to the red and blue filters. As a consequence, the green layer of the produced image contains half of the information carried by the signal, and the rest is carried by the red and blue layers. The downside of using CFAs is obviously that the pixels contain information just from one wavelength, and advanced signal processing methods are needed to compensate the aliasing caused by the sampling method. (Ramanath et al., 2005)



G	R	G	R	G	R
B	G	B	G	B	G
G	R	G	R	G	R
B	G	B	G	B	G
G	R	G	R	G	R
B	G	B	G	B	G

Figure 2.2: Structure of a color filter array (Ramanath et al., 2005)

Pre-processing

The raw data from the sensor needs to be further processed in order to produce an accurate or a pleasing representation of the scene. The steps needed in the preprocessing phase are usually (Ramanath et al., 2005):

- *Defective pixel correction*: The sensor manufacturing process can produce defective pixels in the sensors, which will cause visible errors in the final stored image. The defective pixels can be corrected using interpolation techniques applied on the neighbors of the defective pixels.
- *Linearization*: step is needed if the sensor output is logarithmic instead of linear, since the latter stages in the pipeline rely on linear signal. Usually the most common type of sensors, i.e., charge coupled devices (CCD), produce a linear signal and this step is not necessary.
- *Dark current compensation*: The dark currents are caused by thermally generated electrons in the camera sensor. These currents are visible even with the lens cap on. The techniques for compensation of the dark current are based on either measuring the intensities caused by dark currents by placing an opaque mask around the sensor and measuring the intensities under it, or capturing a dark image, i.e. shutter on, before capturing the scene, and subtracting the dark image signal from the captured scene.
- *Flare compensation*: If there is a bright light source in the field of view, it causes the light entering the optics of the camera to scatter and reflect, producing shifts in the measured signal energy. An example of lens flare can be seen in figure 2.3. The flare compensation algorithms are mostly proprietary, but simple techniques exists, e.g. subtracting a fixed percentage of the mean of a signal neighborhood from the values of the neighborhood.

White balance

Due to a phenomenon of color consistency, human visual system has an ability to map white, or whitish, colors to white even when an object has different radiance when illuminated by different light sources. For example, the human visual system sees a white sheet of paper as white both the light source is an incandescent bulb and when the source is fluorescent lighting. *White balance adjustment* is needed in order to allow digital cameras to correctly identify white color and tune the image color balance, There are some techniques, e.g. assuming that the color channels of an image should average to grey, i.e., *grey world assumption*, or assuming that the white color is found at the point where all the color channels reach their maximum. Also, high-end professional equipment usually allows for manually setting the lighting conditions in order to allow some kind of advanced heuristic to compute the white balance. (Ramanath et al., 2005)

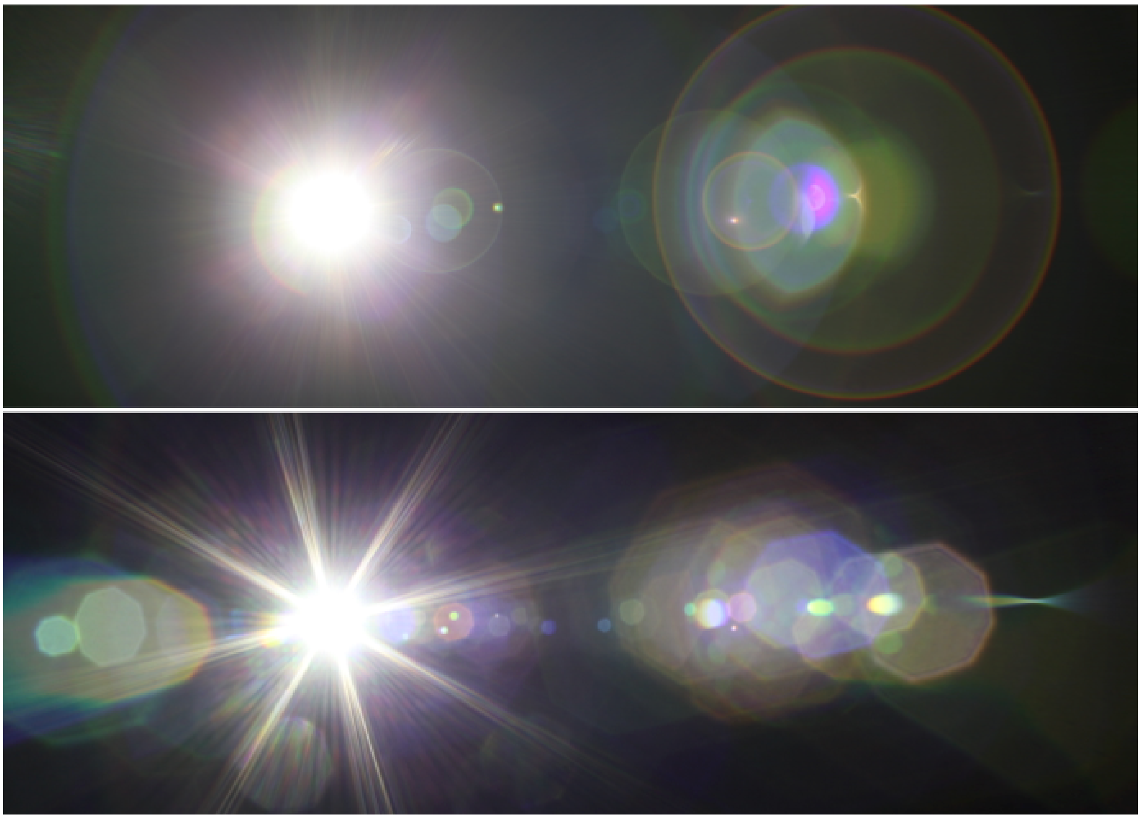


Figure 2.3: A simulated flare effect caused by sun (Hullin et al., 2011)

Demosaicking

As mentioned before, the color filter array allows only certain wavelengths to pass through the filter array. Therefore a pixel of the unprocessed image contains only information from those specific wavelengths. *Demosaicking* techniques are applied in order to estimate the values of the colors that were not measured. (Ramanath et al., 2005)

Feasible demosaicking approaches include heuristics that are based on assumptions on the color images, reconstruction methods that exploit the inter-channel correlations to formulate and solve a mathematical optimization problem, and techniques that utilize the knowledge on the image formation process and try to interpolate the missing pixels as an inversion problem. The demosaicking methods are patented and proprietary, and the exact methods used in real cameras are not known. (Gunturk et al., 2005)

Color transformations

The image needs to be in different formats for the calculations and for output medium. First the raw data is transformed to an *unrendered color space*, calculations are performed, and then the image is transformed into a *rendered color space* for the output medium. The computations include obtaining colorimetric accuracy for the image before transforming the image data to be used for output. (Ramanath et al., 2005)

Post-processing

The aforementioned phases of the imaging pipeline introduce artifacts into the digital image signal. The artifacts have to be removed in the *post-processing* phase before the image is compressed and stored to maintain the high quality. Some of the common post processing phases are color artifact removal, edge enhancement and coring (Ramanath et al., 2005):

- *Color artifact removal*: the demosaicking phase introduces color artifacts, e.g., zipper and confetti, that have to be removed while maintaining the sharpness of the image.
- *Edge enhancement*: in order to produce visually pleasing image, edges in the image have to be enhanced, since human eye prefers clear edges instead of blurred ones. Also, the human eye is less sensitive to the diagonal edges than to vertical and horizontal ones, and even less sensitive to edges in other directions. Edges can be enhanced e.g. by reducing the low frequency content in the image
- *Coring*: images contain data that has no significant contribution to the image detail and behaves like noise. In addition to littering the image, the noise like data increases the size needed to store an image. Coring methods are used to remove such data, e.g. by using transformations on various levels of detail, and then using thresholding techniques on to the transformation coefficients. For example, wavelet transform can be used in this process to analyze the image in desired level of detail.

Compress and store

In *compress and store* phase the image is first transformed into the appropriate color space; the original sensed signal is in additive red, green and blue (RGB) color space which does not necessarily match the intended use of the image. Different mediums require different color spaces: cyan, magenta, yellow and black (CMYK) for reproduction on subtractive color system, e.g. printer, and 8 bit RGB for e.g., display. The image storage and compression scheme has to be selected, for example the Tagged Image FIle Format (TIFF) that can store also the raw data and image parameters, or Joint Photographic Experts Group (JPEG) format which offers effective image compression. Cameras can also store image metadata such as location, camera model and dimensions in Exchangeable Image File Format (EXIF). Finally, the image data is written in a permanent storage such as a flash memory.

The functionality of each phase is summarized in table 2.1.

Table 2.1: The phases of digital camera imaging

Phase	Function
Sensor, aperture and lens	Measuring the color bands, RGB or CMYK
Preprocessing	Linearization, dark current compensation, flare compensation
White balance	White color detection and color space adjustment
Demosaicking	Estimating the undetected pixels using neighborhood information
Color transformations	Obtaining colorimetric accuracy
Post-processing	Removal of artifacts created by the previous phases
Compress and store	Compress image data and store the image in permanent memory

Chapter 3

Sensor pattern noise

As stated in chapter 2, the imaging sensor is a source of noise. Even when pictures are taken in exactly similar conditions in an evenly lit scene, there are small differences in the images produced by two different cameras. Furthermore, the noise is not static, i.e., two consecutive pictures from same camera have slight differences caused by the sensor noise. There are more than one type of sensor pattern noise. As can be seen in figure 3.1, there are two main types: *fixed pattern noise (FPN)* and *photo-response non-uniformity (PRNU)*. PRNU noise divided into two subcategories, i.e., *low-frequency defects* and *pixel non-uniformity noise (PNU)*, the latter being the phenomenon of interest in this study. (Lukáš et al., 2006)

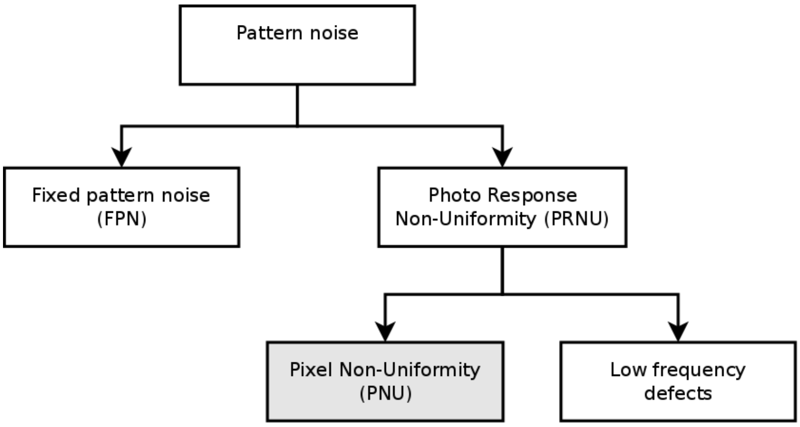

Pattern noise
Fixed pattern noise (FPN)
Photo Response Non-Uniformity (PRNU)
Pixel Non-Uniformity (PNU)
Low frequency defects

Figure 3.1: A classification for sensor pattern noise (Lukáš et al., 2006)

Fixed pattern noise refers to pixel-to-pixel noise caused by dark currents when the sensor array is not exposed to light. *Dark current* refers to the rate of electrons accumulating in each sensor pixel due to thermal action caused by the thermal energy inherent to the structure of the sensor and is independent of light falling on it. (Rocha et al., 2011) Fixed pattern noise is additive, and can be suppressed by automatically subtracting a dark frame from every image they take. It also depends on temperature and exposure. Fixed pattern noise can be relatively easily filtered, and therefore is not of interest in this report.

Photo-response non-uniformity (PRNU) consists of two types of noise: low frequency defects (LFD) and pixel non-uniformity noise (PNU). Light refraction of dust particles and optical surfaces, and zoom settings are the common sources of low-frequency defects. Low-frequency defects are inherently not a characteristic of the imaging sensor, and therefore are not useful for camera source identification - after all, dust particles can be anywhere, independent of which camera is used.

Pixel non-uniformity noise is caused by the different sensitivity of color filter array pixels to light. The source of the varying sensitivity is in the heterogeneity of silicon wafers and imperfections during the manufacturing process. It is unlikely that the sensors would exhibit same correlation of pixel non-uniformity noise patterns, even if they came from the same wafer. Pixel non-uniformity noise is not affected by ambient temperature or humidity. The main reason for concentrating on the PNU noise is that as a pattern noise, it is a deterministic component and present in every photo taken; "noise" in this case does not have the usual connotation of non-determinism or randomness. (Lukáš et al., 2006)

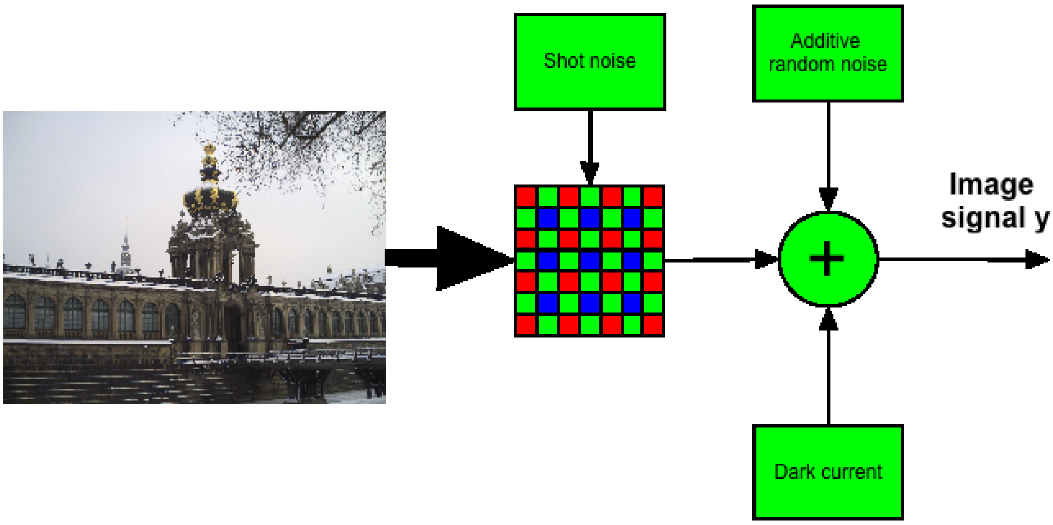


Figure 3.2: Noise model for acquiring the image signal

Figure 3.2 shows the noise model for acquiring the image signal for the image processing pipeline shown in figure 2.1. These types of noise are caused by the imaging process, as explained above. *Additive random noise* is caused by e.g. reading image from the sensor; *shot noise* refers to the random effects of photons on the imaging sensor on the moment the shot is taken. A mathematical model explaining how noise affects the image signal can be formulated as: (Lukáš et al., 2006)

$$y_{ij} = f_{ij} \left(x_{ij} + \eta_{ij} \right) + c_{ij} + \epsilon_{ij}$$

Here y_{ij} is the image signal pixel value, (i, j) is the pixel location, η_{ij} is the *shot noise*, c_{ij} is the noise caused by dark currents, i.e., *fixed pattern noise*, and ϵ_{ij} is the additive noise from various sources. The point of interest are the factors f_{ij} typically very close to 1, as they capture the PRNU noise, which in turn is multiplicative by definition.

Mathematical model for the final pixel value in the photo can be presented as follows:

$$p_{ij} = P(y_{ij}, N(y_ij), i, j)$$

Here p_{ij} is the final pixel value, P is a nonlinear function of y_{ij}representing the effect of the image processing pipeline, $N(i, j)$ represents the pixel neighbourhood and (i, j) the pixel location.

It can be seen that the problem of extracting pattern noise from the image signal is not a straightforward one, due to the non-linearities and different types of noise. The main point of interest are of course the factors f_{ij}, but they cannot be calculated directly.

Chapter 4

Image filtering methods

This chapter introduces filters that can be utilized in extraction of the fingerprints from images. First, the two filters used in this thesis are presented: gaussian (chapter 4.1 and wavelet (chapter 4.2) filters. Then, other possibilities are briefly explored in chapter 4.3. Mathematical derivations or proofs are not given here, since the thesis is concentrated on practical application of filters, not on theory behind signal processing.

An important question with filtering is how to handle the image boundaries. Since in this thesis only the 1024x1024 area from the center of the green layer is used, the filtered images are created using 1040x1040 area from the center, and then the filtered image is cropped to the desired size. Also, it is required that the images are all of the same size to avoid the artifacts and noise data loss caused by resizing the images. Studying different size of images and spatial transforms, e.g., rotation and warps, is not, due to its overwhelming complexity, a topic for this thesis. However, it can be expected that use of neural networks in combination with wavelets would give outstanding results at some point.

4.1 Gaussian filtering

Gaussian filter belongs to a class of convolution filters. Gaussian filter can effectively remove additive noise from the image, e.g. impulse noise and gaussian noise. (Frery and Perciano, 2013, p. 64)

Convolution of two matrices $g = f * m$ is defined as

$$g(i,j) = \sum_{-\frac{\ell-1}{2} \leq \imath',\jmath' \leq -\frac{\ell-1}{2}} f(i-\imath', j-\jmath')m(\imath',\jmath')$$

In effect, this means that the mask is slid over the image, i.e., *applied pixelwise* to each pixel of the image. The filter considers a $\ell x \ell$ neighborhood of each pixel in order to calculate the final value of the pixel in the filtered image. The mask remains constant, i.e., it does not change during the filtering process.

The gaussian filter mask is computed as follows (Frery and Perciano, 2013, p. 64):

$$m'(i,j) = e^{-\frac{1}{2\sigma^2}(i^2+j^2)}$$

Two parameters are needed:

- The standard deviation σ
- The length of the side of the kernel ℓ

Kernel side length ℓ has strong effect on the efficiency of the filter. The larger the side length is, the stronger is the noise removal. Unfortunately, with a gaussian filter, larger kernel side lengths cause also severe amount of blurring. In this thesis, side length of 3 pixels is used, since the image does not need restoration as such, but only invisible noise removal.

Standard deviation parameter σ controls how rapidly the mask weights decrease; the larger σ is, the slower is the decrease. It should also be noted that the mask weights decrease exponentially. When $\sigma \to 0$, the gaussian filter mask converges to identity mask **I**. Also, when $\sigma \to \infty$, the mask converges to the mean of all elements over the mask. The standard deviation used in this thesis is $\sigma = 0.5$, the default for MATLAB `fspecial` function.

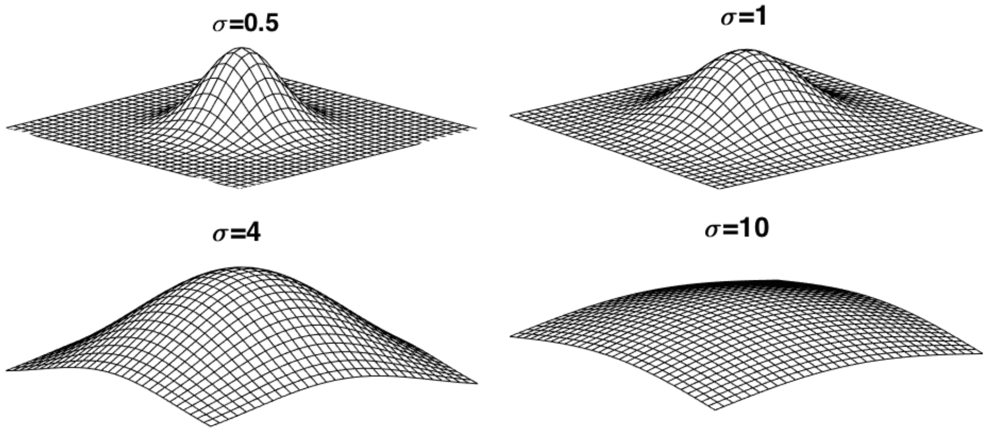


Figure 4.1: Slope of gaussian filter with respect to σ

Figure 4.1 shows the slope of gaussian filter mask for some typical values of σ. As can be seen in the figure, a small value $\sigma = 1$ causes a pronounced spike and steep slope in the mask, while a large value $\sigma = 10$ means gradual slope and no spikes.

As an example, gaussian filters with different parameters are applied on an image of 5 euro bank note in figure 4.2. Figure 4.3 shows a detail of the bank note filtered with gaussian filters. Rows correspond to kernel length $\ell \in \{3, 5, 7\}$, and columns to $\sigma \in \{0.5, 20, 40\}$. As can be seen, the blurriness increases when ℓ grows, and also when σ grows – blurriness increases from left to right and top to bottom.

On the one hand, gaussian filtering is not very accurate and can cause blurriness, but on the other hand it is convolution filters utilizing filtering masks are computationally very fast and easy to implement. Therefore gaussian filtering was chosen as one of the two methods in this thesis, since the idea is to keep the system as simple as possible while maintaining reasonable accuracy.

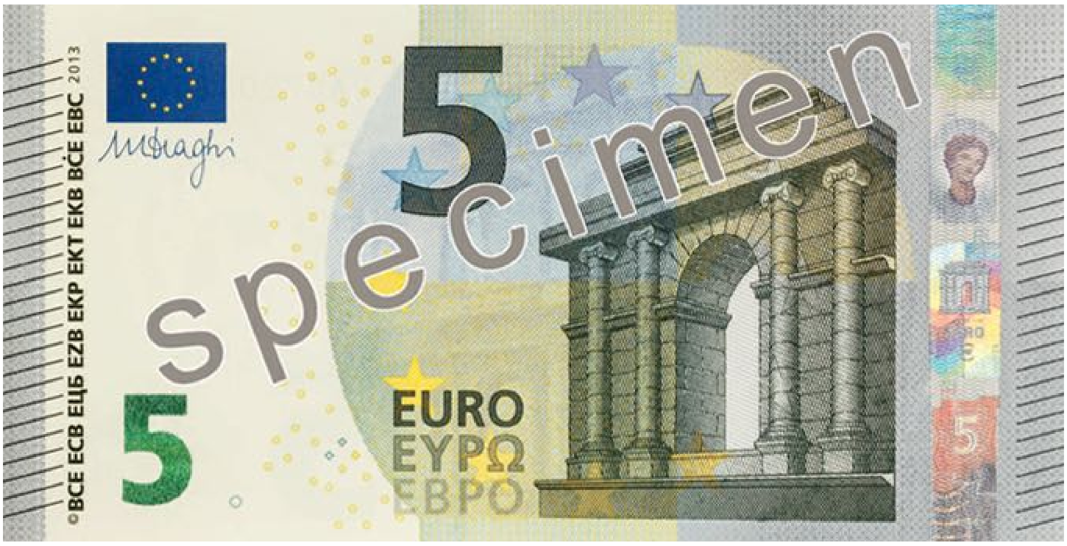


Figure 4.2: The original banknote courtesy of (European Central Bank, 2013)

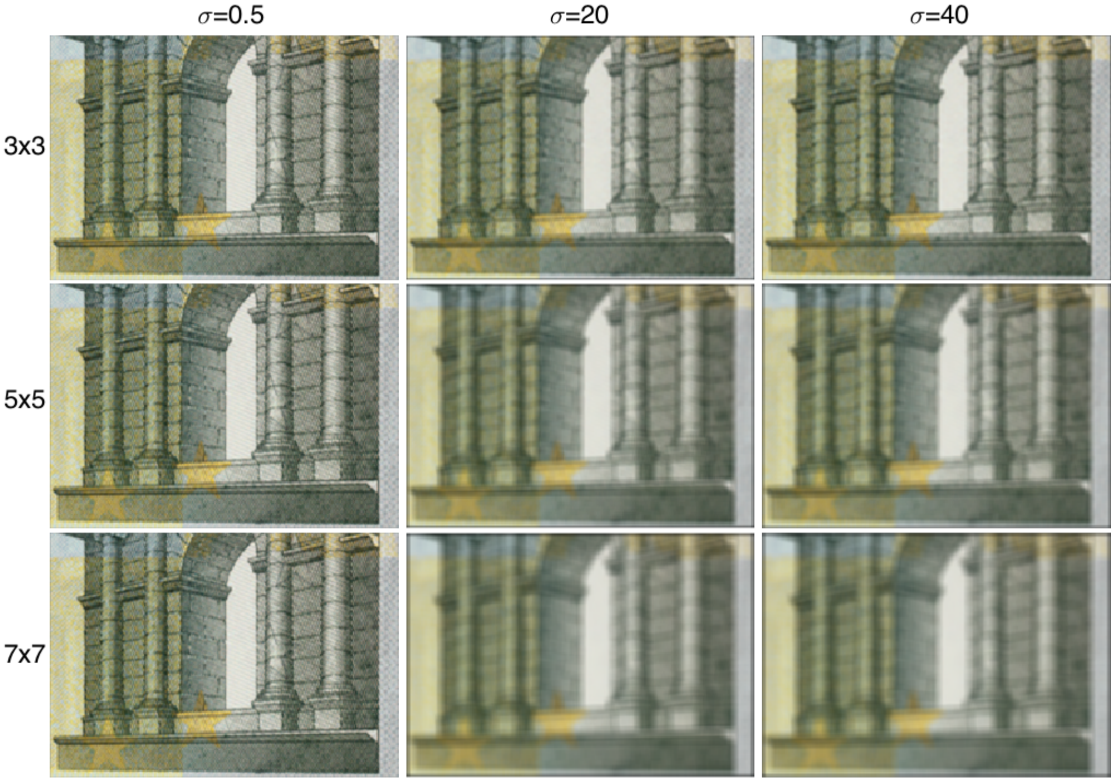


Figure 4.3: Detail from the 5 euro banknote filtered with a gaussian filter

4.2 Wavelet filtering

As was stated in the previous chapter, gaussian filtering has its shortcomings with respect to accurateness of results, namely the blurring it causes. Therefore, a more effective filtering method is needed. The major shortcoming of Fourier analysis is the lack of time-frequency resolution. To amend this lack of more accurate mathematica tools, wavelet analysis was created. The idea of wavelets originated in 1980's, when geologists and physicists needed to analyze complex seismological signals. In 1990's, multi resolution analysis grew into an active field with copious amounts of research on applications, methods and implementation. (Thuillard, 2001)

Wavelets are efficient, even though a bit complex method for signal denoising, compression and zooming. The mathematical derivation is based on Fourier analysis, but wavelets allow for multi resolution analysis of signals, i.e., the signal can be examined in various levels of detail, by scaling and dilating wavelet bases until they fit and cover the signal, and represent it on various levels of detail. Figure 4.4 shows conceptually how wavelet filtering operates on many levels of detail of the signal.

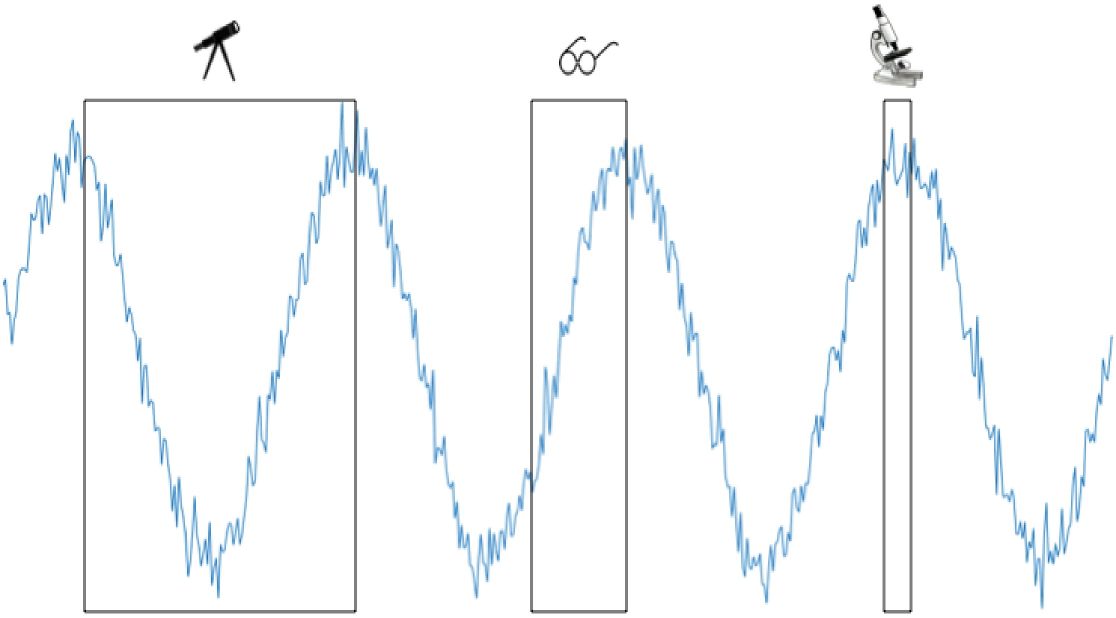


Figure 4.4: Wavelet filtering operates on many levels of detail (Thuillard, 2001)

In order to be a wavelet, function $f(x)$ must fulfill three conditions (Mallat and Mallat, 1999) (Thuillard, 2001):

1. A wavelet function $f(x)$ must average to zero, i.e.,

$$\int_{-\infty}^{\infty} f(x)dx = 0$$

2. A wavelet function $f(x)$ localized, i.e., $f(x)$ is nonzero only in localized area, and zero otherwise, or more formally, there exist a and b such as

$$\begin{cases} f(x) = 0 & x < a \vee x > a \\ f(x) \neq 0 & \text{for most } a \leq x \leq b \end{cases}$$

3. A wavelet function $f(x)$ is oscillating

As can be seen, condition 3 follows from condition 1, since without oscillations condition 1 would never be true, and condition 2 prevents the $f(x)$ from being all-zero. Since wavelet functions are dilated and scaled, non-local wavelets are not useful either.

A wavelet base at minimum comprises two functions: a mother wavelet function ψ, and a scaling function ϕ. The signal is transformed to wavelet domain by using discrete wavelet transform (DWT). DWT is performed by using the wavelet base functions to create discrete convolution linear time-invariant (LTI) highpass and lowpass analysis and synthesis filters. The analysis filters are then applied on the discrete signal to get wavelet coefficients, the coefficients are operated on, and then the signal is reconstructed with inverse discrete wavelet transform (IDWT) by using highpass and lowpass analysis and synthesis filters. When used in practice, wavelets can be divided into two groups according to how the filters are formed, i.e., orthogonal and biorthogonal:

1. Orthogonal wavelet bases utilize only the ϕ and ψ functions to form both the analysis and synthesis filters.

2. Biorthogonal wavelet bases utilize two functions: decomposition scaling ϕ_d and mother wavelet ψ_d functions for analysis filters, and reconstruction scaling ϕ_r and mother wavelet ψ_r functions for synthesis filters. (Tang, 2009)

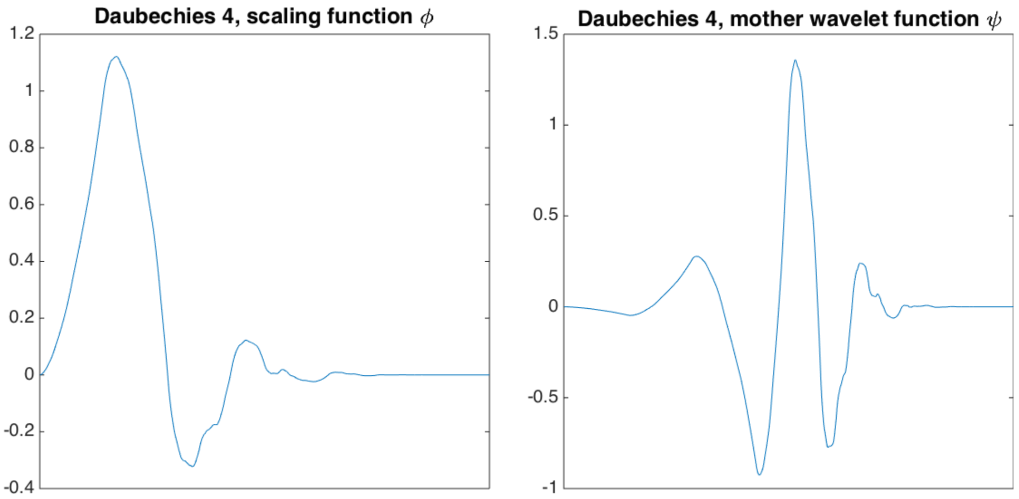


Figure 4.5: Wavelet bases: Daubechies 4 scaling ϕ and and mother wavelet ψ functions

Wavelet bases are usually a family of wavelets defined by some parameters. For example, the figures 4.5 and 4.6 show the graphs for Daubechies family wavelet base with parameter 4. Note that both the synthesis and analysis filters are derived from the same functions, since Daubechies bases are orthogonal.

A mathematical introduction to wavelets and wavelet families can be found e.g. in (Mallat and Mallat, 1999) (Thuillard, 2001) (Tang, 2009) (Mojsilović et al., 2000)

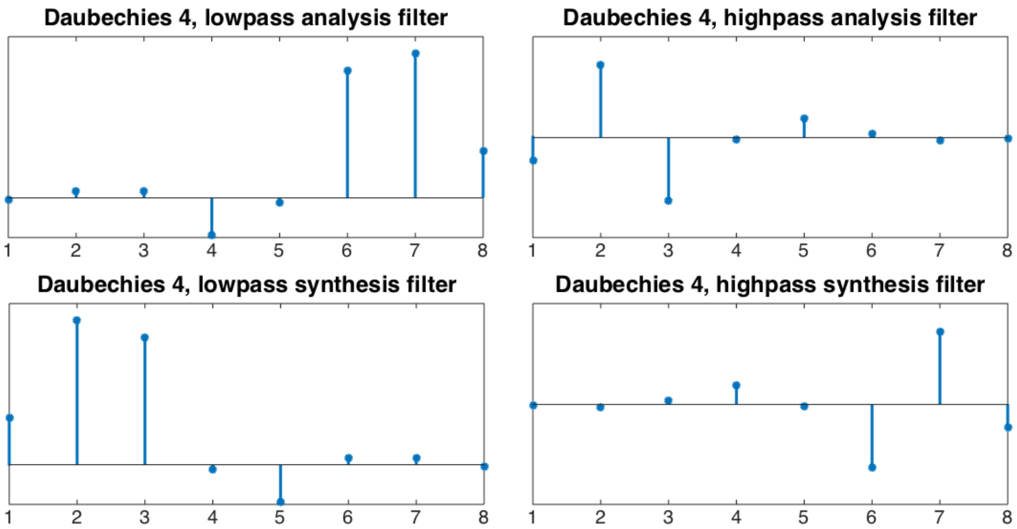


Figure 4.6: Wavelet bases: Daubechies 4 analysis and synthesis filters

The filtering process

Wavelet filtering process has four steps, that are examined below:

1. Wavelet base selection
2. Signal analysis, discrete wavelet transform (DWT), and downsampling
3. Coefficient thresholding
4. Signal synthesis, inverse discrete wavelet transform (IDWT), and upsampling

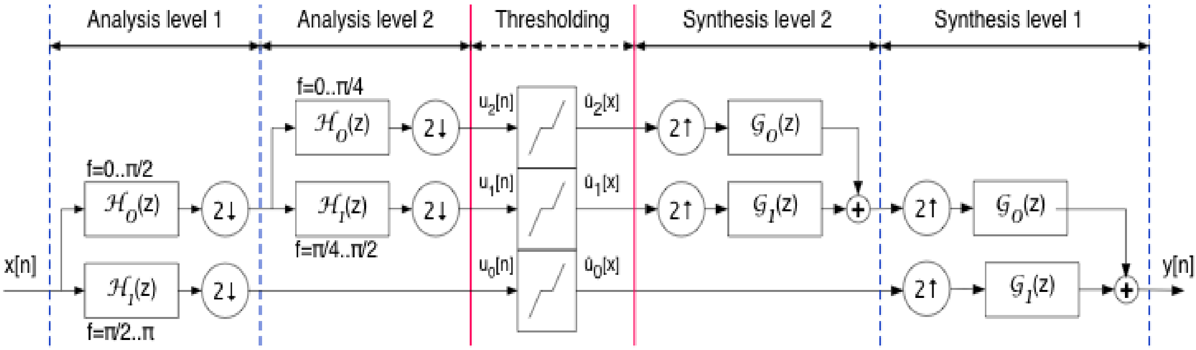


Figure 4.7: Wavelet denoising, big picture (Mitra, 2011, pp.846–848)

Figure 4.7 shows the denoising process. The left part of picture shows the analysis phase with filtering and downsampling. The center part shows wavelet thresholding, in this case the icon shows *soft* thresholding. Finally, the right part shows synthesis filtering and upsampling.

Wavelet base selection

There are no general instructions for selecting wavelets. Instead, the criteria depend on the application area, e.g. electrocardiogram (ECG) denoising (Tan et al., 2007), satellite image compression (Memane and Ruikar, 2014), texture characterization (Mojsilović et al., 2000) and medical image enhancement (Tsai et al., 2002).

Finding the correct wavelet to be used is not a straightforward task. In this thesis, wavelet base 2.4 from biorthogonal spline family was chosen, since an educated guess is that the image filtering for camera identification could be close to texture pattern recognition where the wavelet base performed well enough. (Mojsilović et al., 2000, p.2049)

Signal analysis

Signal analysis phase applies lowpass $\mathcal{H}_0(z)$ and highpass $\mathcal{H}_1(z)$ *analysis* filters recursively to transform the signal $x[n]$ to wavelet domain coefficients $u_k[n]$, where k is the level of detail. Due to the discrete nature of images, the whole analysis process is designated as discrete wavelet transform (DWT).

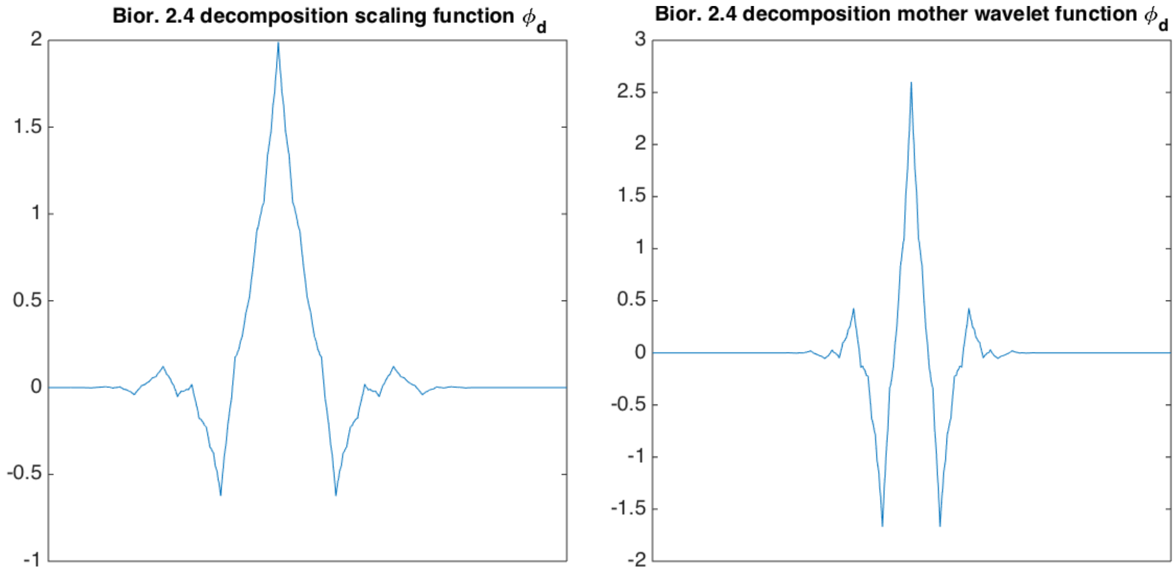


Figure 4.8: Biorthogonal 2.4 decomposition scaling ϕ_d and mother wavelet functions ψ_d

Since biorthogonal 2.4 wavelet base was chosen, the ϕ_d and ψ_d functions in figure 4.8 are needed in the analysis phase. The first task in the analysis phase is to form the $\mathcal{H}_0(z)$ and $\mathcal{H}_1(z)$ filters from the wavelet. The filters are formed from discretization of the continuous wavelet base, but the exact mathematical proof is not a topic for this thesis. As stated before, these filters are discrete linear time-invariant (LTI), and can be seen in figure 4.9.

The filter bank for the analysis phase can be seen in figure 4.10. Analysis phase assumes that most of the information is found in the low frequencies. Therefore, at each level, the output of the highpass filter $\mathcal{H}_1(z)$ is downsampled by factor 2, and output as wavelet coefficients $u_{k-1}[n]$. The output of lowpass filter $\mathcal{H}_0(z)$ is downsampled by factor 2, and output to next level. Downsampling allows for splitting the low frequency of the signal so that the filters can access the respective frequency zones. Finally, on the last level, the output from the lowpass filter is output as coefficients $u_{k_f}[n]$, where k_f is the number of the last level.

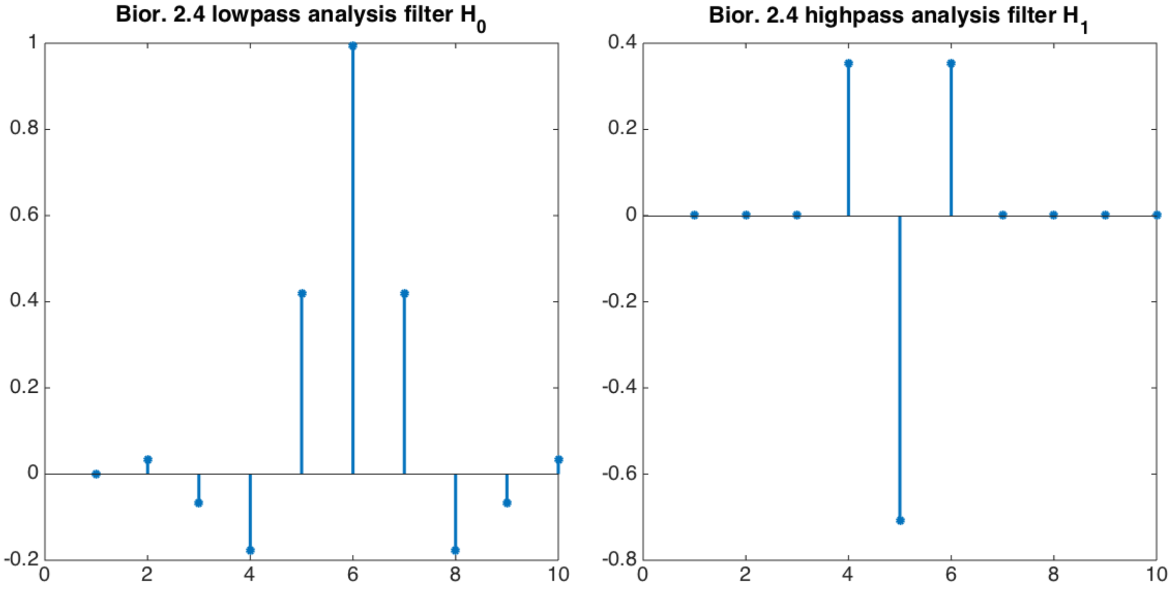


Figure 4.9: Biorthogonal 2.4 analysis lowpass $\mathcal{H}_0$ and highpass $\mathcal{H}_1$ filters

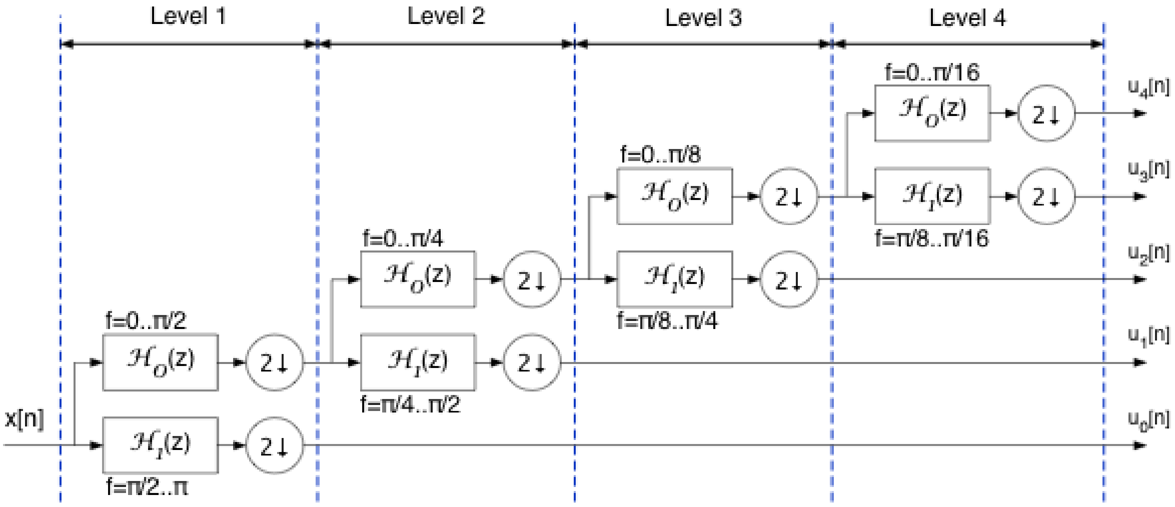


Figure 4.10: Schematic for discrete wavelet transform (DWT), using analysis lowpass $\mathcal{H}_0(z)$ and highpass $\mathcal{H}_1(z)$ filters (Mitra, 2011, p.846)

Wavelet coefficient thresholding

The idea of coefficient thresholding is setting the smallest, i.e., closest to zero, coefficients to zero, to eliminate variations caused by noise. This implies that the signal is approximated by only the larger coefficients. The problem is to set the correct threshold to avoid losing information or reconstruction of noise. The thresholding process can be seen in the middle part of on figure 4.7. Thresholding takes the coefficients $u_k[x]$ from the analysis phase, and produces thresholded coefficients $\hat{u_k}[x]$ to be input to synthesis phase.

There are two methods for coefficient thresholding: soft thresholding zeroes out coefficients that are near to zero, and transposes the rest coefficients so that the curve is continuous. Hard thresholding does not transpose the non-zero coefficients, and thus there are two discontinuities just before and just after the zeroed part, i.e., curve is only piecewise continuous.

Mathematically adding term $\pm T$ takes care of the transposing coefficient values, i.e., smoothing the curve, in soft thresholding equation (Mallat and Mallat, 1999)

$$\hat{u}_k[n] = \begin{cases} u_k[n] + T & \text{if } u_k[n] < -T \\ 0 & \text{if } |u_k[n]| \leq T \\ u_k[n] - T & \text{if } u_k[n] > T \end{cases}$$

There is no similar term $\pm T$ in the hard thresholding equation, and therefore the coefficients are not transposed (Mallat and Mallat, 1999):

$$\hat{u}_k[n] = \begin{cases} u_k[n] & \text{if } |u_k[n]| \leq T \\ 0 & \text{otherwise} \end{cases}$$

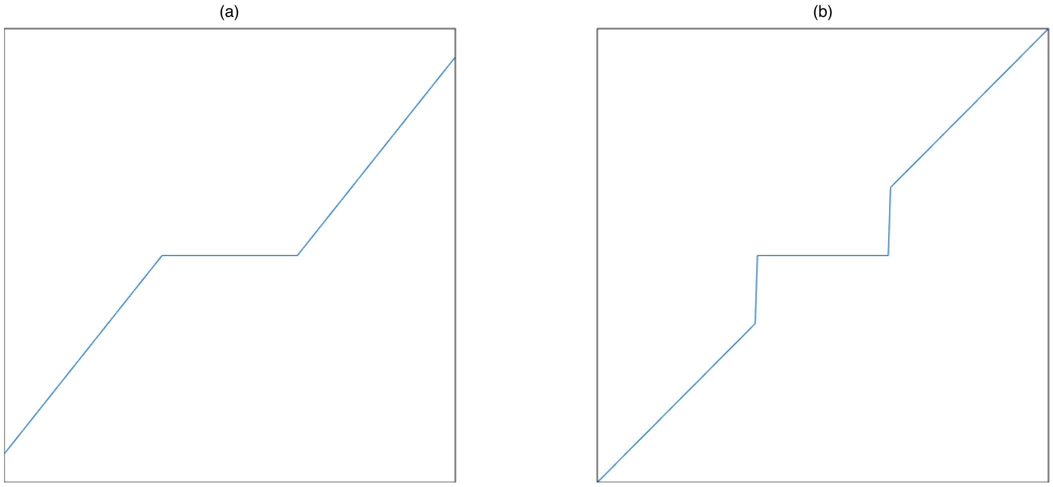


Figure 4.11: Soft (a) and hard (b) thresholding

Figure 4.11 shows a visualization of the soft (a) and hard (b) thresholding curves. It can be seen how coefficients are zeroed and the curve smoothed by transposing larger coefficients, when soft thresholding is applied. As can be seen, the hard thresholding curve zeroes the smaller coefficients, but the larger coefficients are not transposed, and therefore there is a discontinuity, or a jump, before and after the thresholding value.

Choosing between hard and soft thresholding is not straightforward either. A certain type of thresholding can be best of for some application, and almost useless for another. Comparing the results is not straightforward either, and in case of e.g. photos, the "best" result is a topic of subjective perception. For choosing the exact value of the threshold, there are some algorithms, see e.g. (Donoho, 1995) (Thuillard, 2001).

Signal synthesis

Signal synthesis, a.k.a. reconstruction, is operation that applies inverse discrete wavelet transform (IDWT) on the thresholded wavelet coefficients in order to recover the denoised signal. To be exact, IDWT is performed by convoluting synthesis lowpass and highpass filters $\mathcal{G}_0$ and $\mathcal{G}_1$, respetively, over the thresholded wavelet coefficients $\hat{u}_k[n]$.

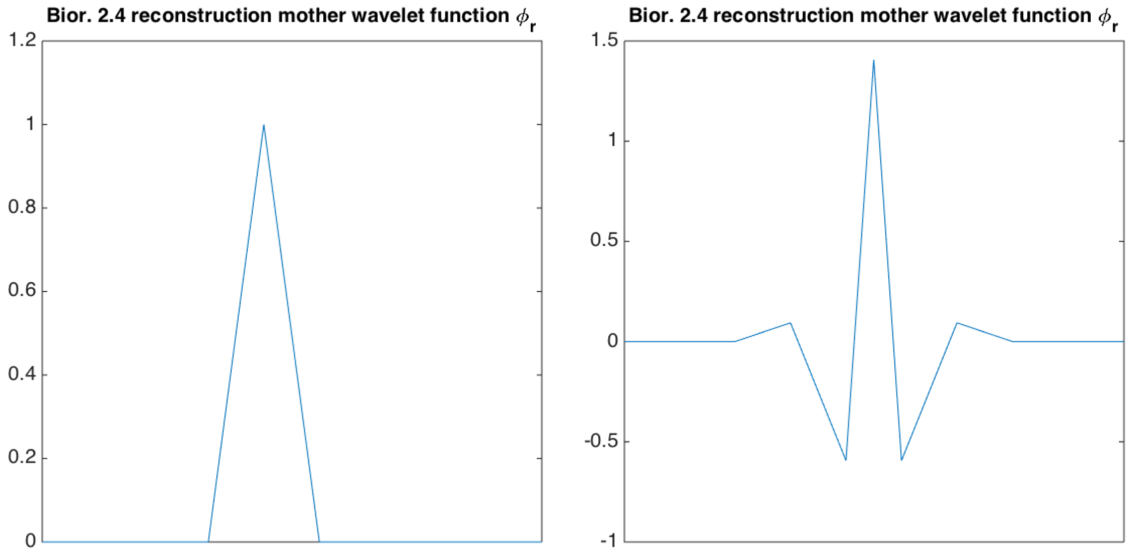


Figure 4.12: Biorthogonal 2.4 reconstruction scaling ϕ_r and mother wavelet functions ψ_r

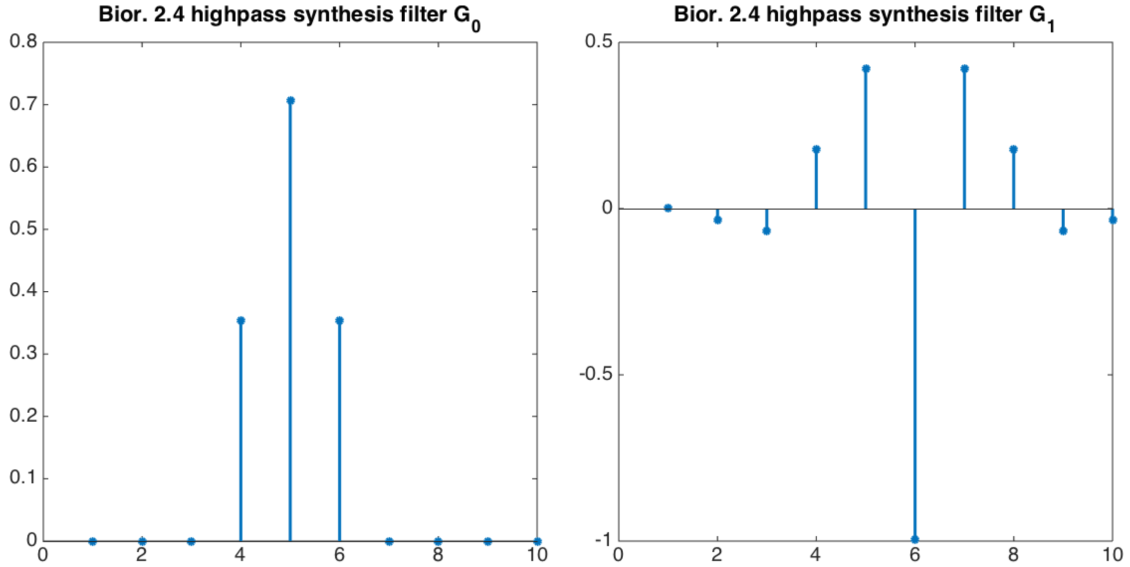


Figure 4.13: Biorthogonal 2.4 synthesis lowpass $\mathcal{G}_0$ and highpass $\mathcal{G}_1$ filters

Figure 4.12 shows the wavelet reconstruction scaling ϕ_r and mother wavelet function ψ_r for biorthogonal 2.4 wavelets. As stated before, orthogonal wavelet bases do not need separate decomposition and reconstruction functions, since the synthesis filters can be derived from the same base as the analysis filters. Figure 4.13 shows the synthesis filters derived from the reconstruction functions.

In the case of this thesis, the lowpass and highpass synthesis filters $\mathcal{G}_0$ and $\mathcal{G}_1$ respectively, are derived from the biorthogonal 2.4 wavelet reconstruction functions. As stated before, the filters are applied on, i.e., convoluted over, the thresholded

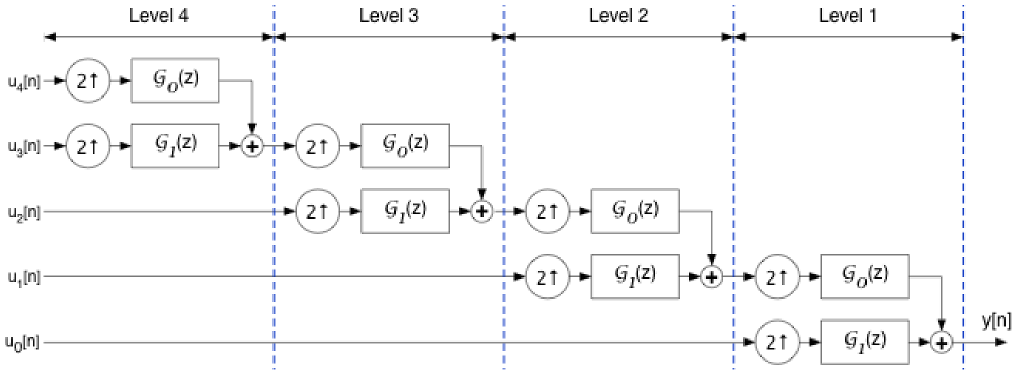


Figure 4.14: Schematic for inverse discrete wavelet transform (IDWT), using analysis lowpass $\mathcal{G}_0(z)$ and highpass $\mathcal{G}_1(z)$ filters (Mitra, 2011, p.846)

wavelet coefficients. The filters are applied recursively, analogously with the signal analysis phase.

Figure 4.14 shows the filter bank that is used in the signal synthesis. The filters are applied in the reverse order of the levels of detail, i.e., the most detailed coefficients first. The first synthesis is performed by first upsampling two coefficient vectors by factor two, and then applying lowpass synthesis filter on the coefficients $\hat{u}_{k_f}[n]$, where k_f is the most detailed level (in the figure 4), and highpass filter on coefficients $\hat{u}_{k_f-1}[n]$. The signals are summed, and forwarded as an input to the level $k_f - 1$. The result signal is then upsampled again, filtered with the lowpass filter and combined with the next upsampled and highpass filtered coefficients. This is continued until the coefficients $\hat{u}_0[n]$ have been processed. When the synthesis process is completed, the reconstructed signal $y[n]$ is a denoised version of the original signal $x[n]$.

As can be seen, the idea behind wavelet filtering might be complex. However, with suitable mathematics software, e.g., MATLAB® or Octave, the implementation is very simple and there are ready-made utility functions.

4.3 Other possibilities

There are numerous other possibilities for choosing the filtering method to acquire scene content and remove noise from images.

Fourier Transform (FT) and Inverse Fourier Transform (IFT) have long been used to denoise signals, especially when there is a need to eliminate specific frequencies, e.g., the 50 Hz hum caused by power lines. As stated before, Fourier Transform (FT) based methods lack the time-frequence resolution, and therefore wavelet based methods perform much better. However, Fourier Transform based methods are very quick to implement due to the existence of Fast Fourier Transform (FFT) and Inverse Fast Fourier Transform (IFFT) to speed up computation significantly, and therefore Fourier Transform based filtering methods could be studied as an alternative, when a huge number of images have to be analyzed.

Evolving non-linear partial differential equation (PDE) based can be used for filtering an image. Non-linear PDEs have numerous possibilities in addition to denoising, e.g., deblurring and edge enhancement. The filtering methods have different requirements and kernel PDEs, and require considerable expertise to be used. (Liu et al., 2011) (Komprobst et al., 1997) These methods can be rather slow if implemented as such, but they can be speeded up with utilization of neural networks to control the parameter evolution, for example Extreme Learning Machine (ELM) to overcome the slowness of gradient descent learning. (Wang et al., 2011) (Huang et al., 2006) However, due to the complexity of the PDE methods and parameter evolution, these approaches will not considered in this thesis.

Chapter 5

PNU based identification of camera sources

Now that digital camera imaging pipeline and sensor noise patterns and image filtering methods have been introduced, it is time to show how a camera fingerprint is created, and how it is utilized for sensor recognition. In this work the focus is on fingerprints generated from pixel non-uniformity noise (PNU), which is currently considered one of the most viable approaches for sensor recognition. (Chen et al., 2008)

When noise patterns are extracted from images, forensic investigators can use these patterns to compare to others and get information about the relation between those images. Noise pattern analysis can be used to (Baar et al., 2012):

1. Determine if an image is made using the suspect camera
2. Determine if a group of pictures is made using the same camera
3. Determine groups of images created using the same camera, from a database of images

This thesis is concentrated on alternatives 1 and 2. Alternative 3 requires clustering methods, since it implies blind source images, i.e., the camera is not known. Alternative 3 cannot be studied until the alternatives 1 and 2 have been thoroughly analyzed.

In these applications, it is necessary to link an image or a video-clip to a specific piece of hardware. Sensor photo-response non-uniformity (PRNU) has been previously proposed (Lukáš et al., 2006) as an equivalent of "biometrics for sensors", especially pixel non-uniformity noise (PNU), which is used in this thesis. There are several important advantages of using this fingerprint for forensic purposes: (Goljan and Fridrich, 2008)

1. Stability. The fingerprint is stable in time and under a wide range of physical conditions.
2. Generality. The fingerprint is present in every picture (with the exception of completely dark images) independently of the camera optics, settings, or the scene content.

3. Universality. Virtually all sensors exhibit PRNU (both CCD and CMOS sensors).

4. Dimensionality. The fingerprint has large information content because it is a signal that contains a stochastic component due to material properties of silicone and the manufacturing process itself. Thus, it is unlikely that two sensors will possess similar fingerprints.

5. Robustness. The fingerprint survives a wide range of common image processing operations, including lossy compression, filtering, and gamma adjustment.

Using photo-response non-uniformity (PRNU) requires synchronization: if the image has been cropped or scaled, PRNU detection will not succeed. (Goljan and Fridrich, 2008)

In order to generate fingerprints from individual images, the first task is to remove the image content from each of the images so that only the noise pattern remains. This is achieved by using efficient filtering techniques such as wavelet filtering, that is also applied in this work. The actual fingerprint is generated by calculating the mean of the noise patterns. Testing whether an image is from a specific camera is done by calculating the correlation between the noise residual from that image, and the camera fingerprint.

The structure of this chapter is as follows:

- First, the method for generating the pixel non-uniformity noise (PNU) based fingerprint is introduced in a more detailed level.
- Second, calculation of the correlations between the fingerprints and the noise residuals is formulated.
- Third, some example data is presented.
- Fourth, principal component analysis is described as a method to reduce dimensionality of the correlation data.

This chapter gives only a rough context for the classification methods, which will be presented in chapter 6. This chapter and the method presented here is based on the approach presented by Lukas et al. (Lukáš et al., 2006).

5.1 Generating PNU based reference patterns

As stated in chapter 3, the source of noise and its effect on the image signal is a complex issue. One way to circumvent the problem of directly estimating the pixel non-uniformity noise pattern is to use camera reference patterns. The idea behind generation of a PNU reference pattern for a camera is simple: a certain amount, usually $N \geq 50$, training images from the camera are processed one by one and then their mean is calculated to form a reference pattern as shown in figure 5.1.

The process can be described as follows:

1. A wavelet denoising filter, or any low frequency cancelling filter, is applied on an image.

2. The denoised image is subtracted from the original to create a noise residual matrix containing only the high frequency noise.
3. The mean of the high frequency noise matrices is calculated to form the reference pattern.

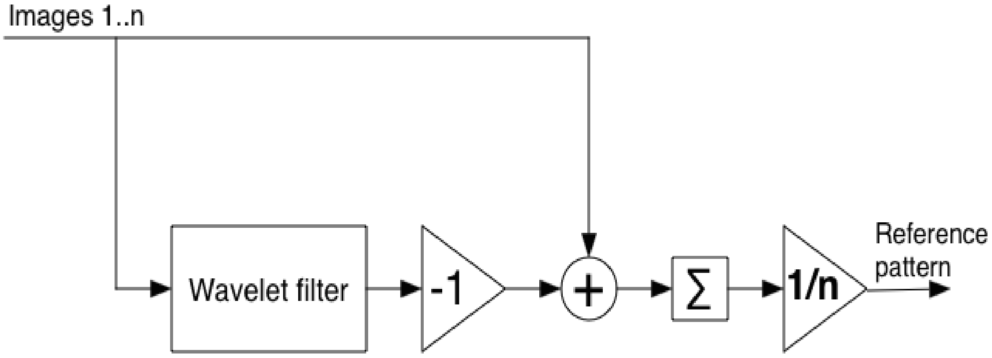


Figure 5.1: Computing the reference pattern

Mathematically the idea is straightforward: the reference pattern $\mathbf{W}^C_{ref}$ for camera C is acquired by averaging noise residuals of the images from that camera. Noise residuals are used to prevent the image content from affecting the reference pattern. The image content is removed by computing the difference $\mathbf{W}_k$ by subtracting the image filtered by filter F from the original image.

$$\mathbf{W}_k = \mathbf{I}_k - F\left(\mathbf{I}_k\right)$$

The reference patterns are then formed by averaging the N training images from the camera. The rest of the images can be used for validation and testing phases of the PNU source camera identification. It was shown in the experiments that N=60 is sufficient for reliable camera identification.

$$\mathbf{W}^C_{ref} = \frac{1}{N}\sum_{i=1}^{m}\mathbf{W}^C_i;\ k \leq N$$

It can be seen from the equations that creation of reference patterns contains highly independent processes, and therefore can be easily modified for parallel computing schemes.

Figure 5.2 shows the schematic diagram of creating referencce patterns. As stated before, the images are first filtered, and then the noise patterns are calculated

Figure 5.3 shows an example of a camera reference pattern. As can be seen, it is not immediately clear that there are specific shapes, but for a computer algorithm, the minor variances between pixels give sufficient information for image classification.

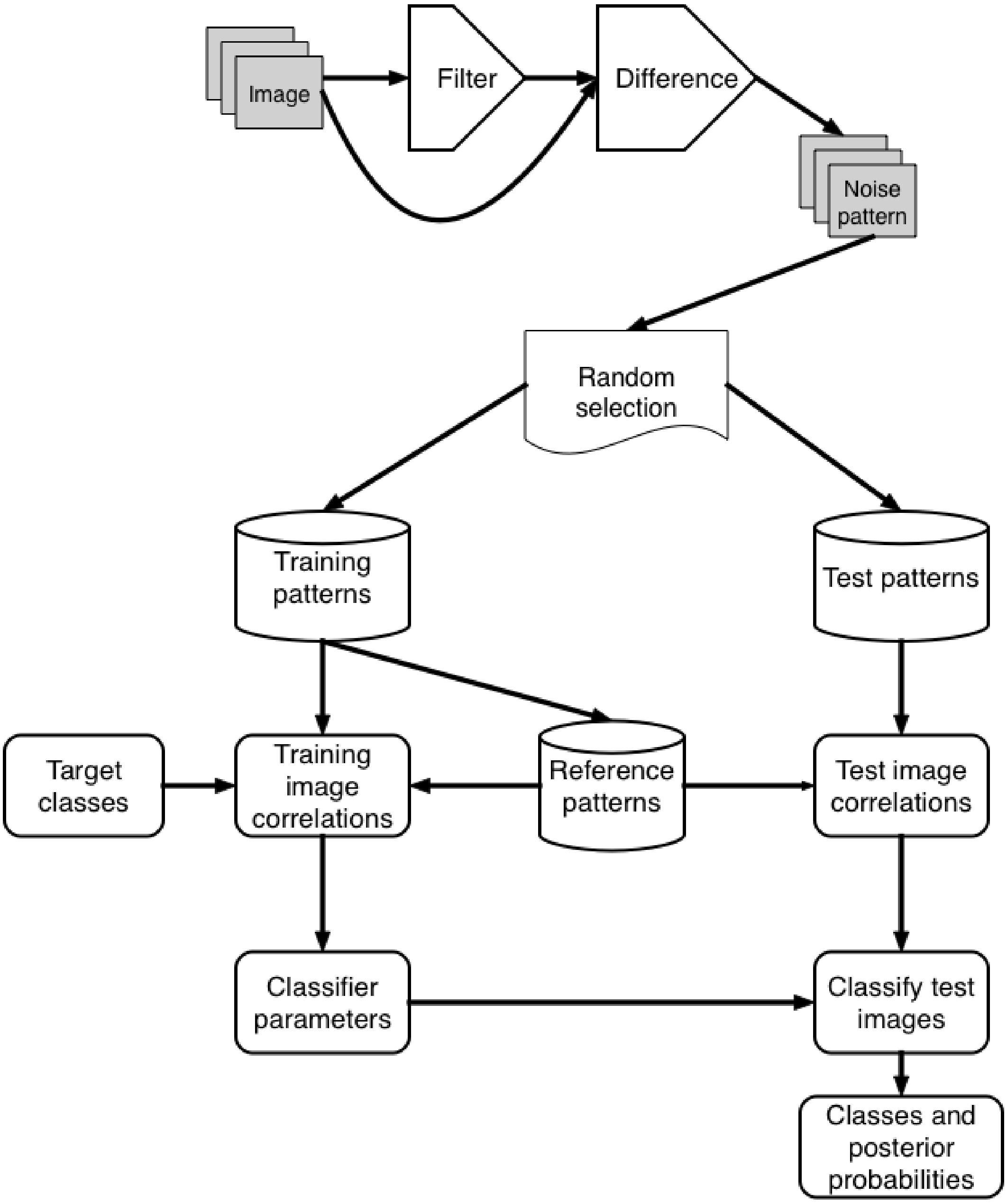


Figure 5.2: Schematic diagram of the recognition system

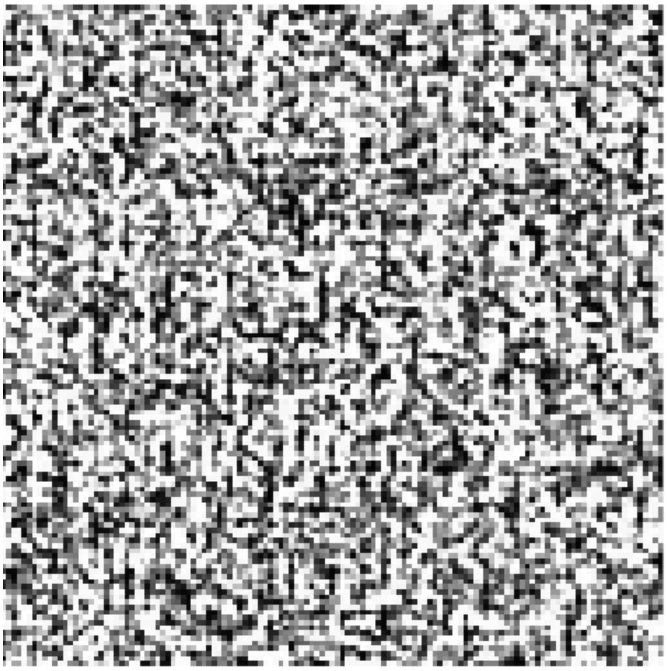


Figure 5.3: Pseudocolor image of a part of a reference pattern

5.2 Correlation of an image and the reference pattern

The correlation ρ^C between the reference pattern $\mathbf{W}^C_{ref}$ of camera C and the kth image $\mathbf{I}_k$ is defined as

$$\rho^C(\mathbf{I}_k) = \frac{(\mathbf{W}_k - \bar{\mathbf{W}}_k) \odot (\mathbf{W}^C_{ref} - \bar{\mathbf{W}}^C_{ref})}{\|\mathbf{W}_k - \bar{\mathbf{W}}_k\| \|\mathbf{W}^C_{ref} - \bar{\mathbf{W}}^C_{ref}\|}$$

where $\bar{\mathbf{W}}_k$ is the mean value of $\mathbf{W}_k$, $\bar{\mathbf{W}}^C_{ref}$ is the mean value of $\mathbf{W}^C_{ref}$, $\alpha \odot \beta$ is the dot product of the matrices α and β, and $\| \cdot \|$ is the matrix norm. (Lukáš et al., 2006)

The main problem with this method comes from the spatial nature: the images should be all taken with the same resolution, or be resized to the same resolution. However, the algorithm is quite robust and simple linear pixel interpolation is enough when resizing. Another important problem are the effects of geometric transformations on the reference pattern matching. It is possible to search for the geometric transformations and invert the transform, or transform the signal itself for example with Fourier-Mellin transform. (Goljan and Fridrich, 2008)

It is possible that PNU patterns are not sensitive enough to make the distinction between different cameras of the same model. Recognizing the camera model is reliable, though. Methods for comparing the relative differences between the correlations of the different cameras of the same model are needed.

5.3 Example data

After the correlations ρ_i, $i = 1 \ldots M$ have been calculated for all the M test images, the images need to be assigned to the cameras. Here, the correlations with training and test data sets are demonstrated with example data from our initial experimentations made in this research. Table 5.1 shows a snippet of the produced data. The images are on the rows $1 \ldots N$ and the cameras are on the columns $1 \ldots M$. The highest correlations are marked with bold text.

Table 5.1: Example correlations

Image	Camera 1	Camera 2	Camera 3	...	Camera M
1	**0.1232**	0.0232	0.0012	...	0.00012
2	0.0001	0.0033	**0.1002**	...	0.012
3	0.0502	0.0422	**0.1740**	...	0.0020
...	...	...	...	...	...
N	0.0099	0.029	0.0017	...	**0.1750**

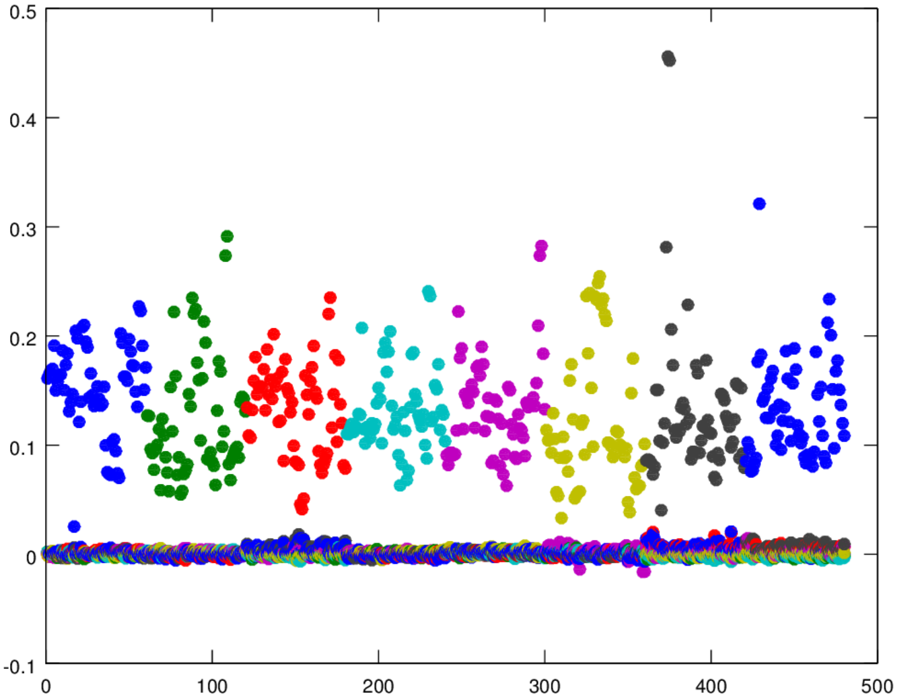


Figure 5.4: Correlations with the training data set

Figure 5.4 shows an example of correlations with the training data. There are seven different cameras marked with different colors. As can be seen in the figure, the difference between matching and non-matching images is quite clear.

Figure 5.5 shows the correlations with the test data. As can be seen, the correlations with the training data are overly optimistic with respect to correlations with the test data. As mentioned in section 6, the difference between $\log_{10}$ correlations of matching and non-matching images should be over 1 in order to maintain high

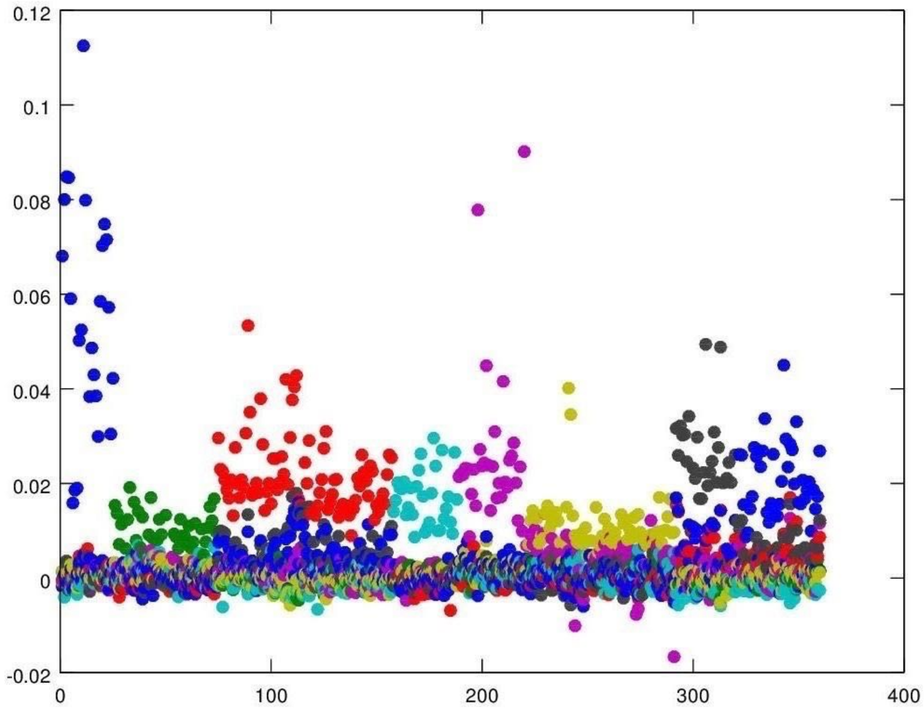


Figure 5.5: Correlations with the test set

accuracy. As can be seen in the figure, the lowest correlations of matching images are quite near to the correlations between the camera and non-matching images.

5.4 Data preprocessing - principal component analysis

The complexity of most of the machine learning algorithms directly depends on the dimensionality of the data. Therefore, it is useful to reduce the dimensionality of the data without losing important information. Principal component analysis does exactly this – the features are projected on a new feature space with a smaller number of components than in the original, while preserving as much variance as possible. Principal component analysis is connected to factor analysis – while principal component analysis aims to preserve the variance of existing components, factor analysis assumes that there are latent variables that can characterize the dependency between the original features with lower dimensionality. (Alpaydin, 2010, pp. 113–125)

Figure 5.6 shows an example of components created by principal component analysis from a data set with 450 variables . There are many ways to select how many components are included in the data with reduced dimensionality. One way is to select a cutoff point with enough variance explained. For example, the cutoff point can be set so that 90% or 95% variance is explained. In figure 5.6, the cutoff point is fixed to 90%, which means that 80 first components are selected for further processing. There is a data point that shows the number of components and the explained variance, i.e., $N = 80$ and cumulative variance is $\approx$ 90%.

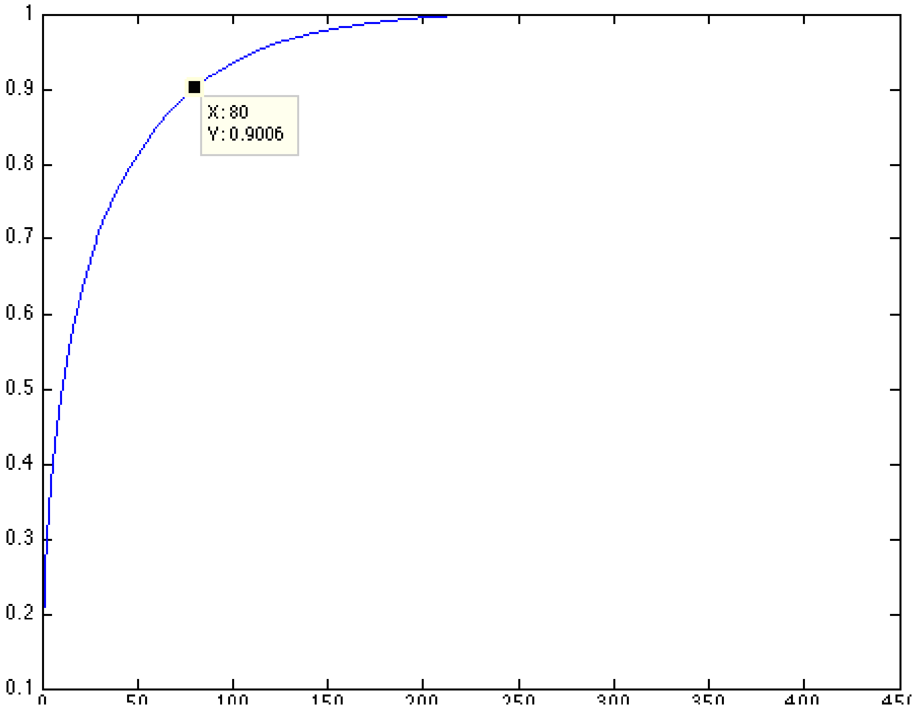


Figure 5.6: Example of principal component analysis

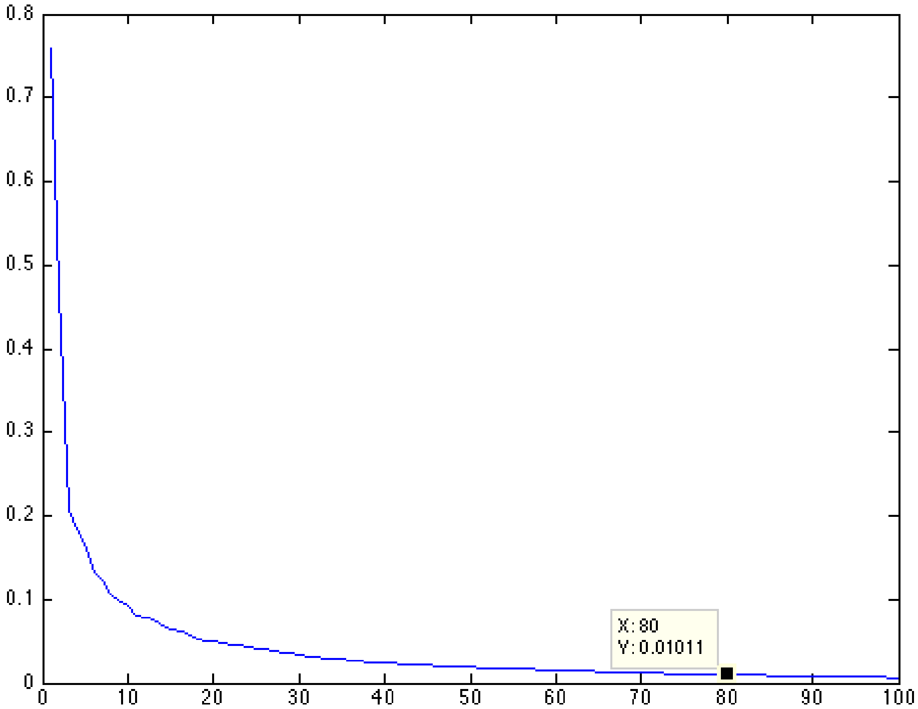


Figure 5.7: Variances of the first 100 components

Figure 5.7 shows the absolute variance of first 100 components. As can be seen in the figure, the variance decreases rapidly with respect to the the index of the component. The components are always presented in decreasing order according to the variance. The data point marked in the figure shows the variance of 80^{th} component, which is very low. It is much easier to find the cutoff point from figure 5.6 showing the cumulative variance, than from absolute variances of figure 5.7.

Chapter 6

Image classification methods

Creating PNU fingerprints and calculating the correlations between image noise residuals and the fingerprints were presented in the previous chapter. It is time now to show how the images can be assigned to camera sources based on the correlations.

In order to choose the classification method it is necessary to first characterize the problem a bit: whether the classification is based on the correlations between one fingerprint and the images, or with more than one fingerprint. Also, it is necessary to decide whether the target is to decide between two questions: “Does the image belong to camera X?” or “To which camera does the image belong?”. Even though the latter alternative sounds tempting since it may appear to solve all the problem at once, it is also more error prone. In this chapter the connotations of those alternatives are discussed in depth.

There are three classification methods presented in this chapter:

1. The correlations are compared naïvely, and the image is assigned to the camera with highest correlation

2. Bayesian classifier can take into account the relationships between the correlations

3. Linear regression that can easily manage varying number of correlation variables, and can easily handle very large sets of correlations

Finally, in this chapter there are some considerations for tuning the algorithm to achieve better performance with respect to memory and processor time consumption.

6.1 Characterization of classification methods

The most important parameter for choosing the classification method is whether there are images from unknown cameras. If it is known that the test images are only from the cameras with known reference pattern, the classification is fairly straightforward with even the most naïve methods, e.g. choosing the class with highest correlation between the reference pattern and the image.

If there are images from unknown cameras, the problem gets a bit harder: there has to be a way to discriminate between the correlations high enough for assigning the image to a camera, and low enough to reject the image. Usually the easiest classification methods do not have a straightforward solution for assigning an image to a "none" or "unknown" class. The problem gets especially hard when the difference between "high enough" and "too low" correlation is small $\leq 10^{-1}$. This usually happens when the amount of noise data is sacrificed for high performance (see chapter 6.5), or when less effective, e.g. gaussian blur, filters are used for low frequency noise filtering.

Since the images forming a reference pattern are from a known camera source, supervised learning methods can be applied. As stated before, there are at least three feasible candidates for the classification of the images:

1. Naïve comparison of correlations (with or without a threshold)
2. Bayesian classification
3. Linear regression

The first method is fairly straightforward and usually fairly reliable. The two latter methods require more rigorous tuning of the algorithms, and can be combined with principal component analysis to reduce the dimensionality of the data, especially when there are a fairly large number of different cameras.

There are two valid approaches when using the classification methods: either the problem is approached as a multi-class classification problem, i.e., the images are directly classified as belonging to one of the cameras, or the problem is reduced to a two class problem. The latter approach requires testing which images belong to the first camera and which do not, then again which of the remaining images belong to the second camera and which do not, and so on. In case of Bayes and linear regression classification, it has to be decided whether an univariate or a multivariate approach is selected, i.e., is the output dependent on all the correlations with reference patterns, or dependent only one of the reference patterns. These considerations are summarized in table 6.1. As can be guessed, using multiclass univariate option does not probably work, since it implies that the correlations between the images and one reference pattern contain enough information to cover all the classes - which is not the case.

One more thing to consider is whether to use principal component analysis (PCA) to reduce the dimensionality of data, when a multivariate method is chosen. If there is a large number of reference pattern, it might be a good idea to use PCA. Sometimes PCA reduces a multivariate problem to an univariate one. This is just a special case of multivariate multi-class problem, as the remaining component carries enough information from the correlations to allow multi-class classification.

There is still one question remaining: how to divide the data into training, validation and test sets. Selection of the training set for reference pattern formation is simple: choose $N \geq 50$ images randomly from the data set, and form a reference pattern. However, there is a catch: when bayesian classification or linear regression are used, there are actually two training phases. The first is the same for all the methods, i.e., creating the reference pattern and then training the classifier. The second phase is either

1. using the same set of images for both creating the reference pattern and training the classifier

Table 6.1: Univariate vs. multivariate, two class vs. multiclass

	Univariate	Multivariate
Twoclass	Decide whether an image is from a certain camera based on the correlation between image and reference pattern of a camera	Decide whether an image is from a certain camera based on the relationships between the correlations of the image with all the reference patterns
Multi-class	*Assign an image to one of the cameras based on the correlations between the image and only one camera*	Assign an image to one of the cameras based on the relationships between the correlations of the image with all the reference patterns

2. using of different sets for creating the reference pattern and for training the classifier

Using two different sets of data is tempting, since the first alternative can easily overestimate the correlation levels between the image and the matching reference pattern. However, using two different sets can easily use up all the images from a camera, and there is no data left for validation and testing. There is no simple answer for this problem, and eventually the choice will be based on the source image database. There is not a large amount of images to choose from, and therefore, in the case of two separate sets there will not be enough data for learning the classes for linear regression and Bayes classifier, and absolutely no data for testing.

A minor detail is that correlations can be both positive and negative. In this report the correlations are compared directly, instead of using absolute values, as the sign is considered being of importance.

6.2 Naïve comparison of correlations

Naïve correlation classifier is essentially a multivariate method with function f producing the class with highest correlation :

$$f(i) = k, \text{ when } \rho_i^{C_k} = \max\{\rho_i^{C_1} \ldots \rho_i^{C_M}\}$$

In other words, the the image is assigned to the first camera that has the highest correlation with the image. When there are more than one cameras with the same correlation, the first is selected. This situation is highly improbable, though, since it essentially requires that two cameras have exactly same reference patterns.

A significant problem with the naïve classification is its total inability to handle images that are from none of the cameras. Naïve comparison will blindly assign the image to one of the known cameras, even though the correlations are significantly smaller than with those images from known cameras.

It is possible to use Neyman-Pearson approach to first determine the distribution of $\rho_C(\mathbf{q})$ for other images taken with camera C, i.e., those that were not used to create the reference pattern. Then the distribution of $\rho_C(\bar{\mathbf{q}})$ is determined, where $\bar{q}$ are the images not taken with camera C. The Neyman-Pearson method allows for minimizing FRR, and imposing bound on the FAR, since in forensic applications FAR should be kept low, for the obvious reason of not making hasty conclusions. (Lukáš et al., 2006)

6.3 Bayesian classification

The simplest classification problem for a Bayesian classifier is one with only one independent variable, or to be exact, a case with only one feature and two classes, namely "image is from camera k" and "image is not from camera k". The problem can be simplified even more by assuming that the correlations follow some known distribution, which implies that the probability density and cumulative density functions are known. In this work the correlations are assumed to be normally distributed due to the nature of many factors affecting the correlations, which in turn implies that the central limit theorem can be more or less safely applied. In addition to the assumption on normality, the variables are assumed to be independent and identically distributed (iid.). The central limit theorem means that the sum of many random variables with unknown distributions asymptotically converges to normal distribution.

In order to classify the samples, the parameters of the distributions have to be estimated. Simplest way is to take test images for a camera, correlate them to the fingerprint of the class, and calculate the mean μ and variance σ^2 for correlations belonging to the class. The mean and variance can be used as parameters for the iid. gaussian distributions of each class.

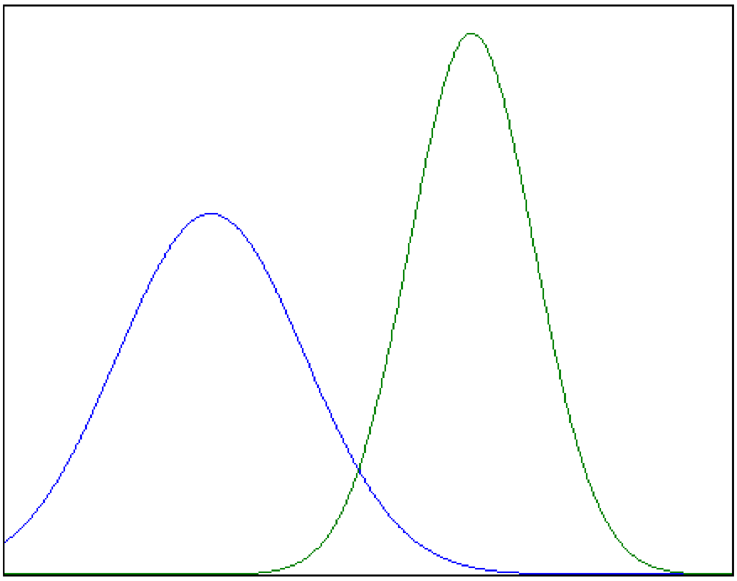


Figure 6.1: Two class univariate classification problem

Figure 6.1 shows an example classification problem with only one feature and two classes. The two peaks, blue and green, represent classes 1 and 2, and correlations are on x axis. As can be seen, the classes partially overlap, i.e., there is a zone of ambiguity in the middle of the figure.

The a posteriori probability for class ω_i for observation x_j can be written (Alpaydin, 2010, p.22) as

$$P(\omega_i|x_j) = \frac{P(x_j|\omega_i)P(\omega_i)}{P(x_j)}$$

It is fairly easy to derive the decision rule for the classification. The principle is simple: choose the class with the highest probability for the observation.

$$\begin{aligned} \frac{P(\omega_1|x_j)P(\omega_1)}{P(x_j)} &\gtrless \frac{P(\omega_2|x_2)P(\omega_2)}{P(x_j)} \\ \Leftrightarrow P(\omega_1|x_j)P(\omega_1) &\gtrless P(\omega_2|x_2)P(\omega_2) \end{aligned}$$

The rest of the cases are explained here conceptually, as the mathematics gets fairly complex, even though the principle remains the same.

The case with multiple variables and two classes is shown in figure 6.2. The distribution is again assumed to be iid. normal. In the figure, the left side shows the multinormal distribution for the two classes (the peaks), and the right side shows contour plot of the two peaks. The simplest rule for choosing one of the classes is again maximizing the posterior probability, i.e., choosing k by rule $\arg\max_{k} \{P(\omega_k|\mathbf{x})\}$.

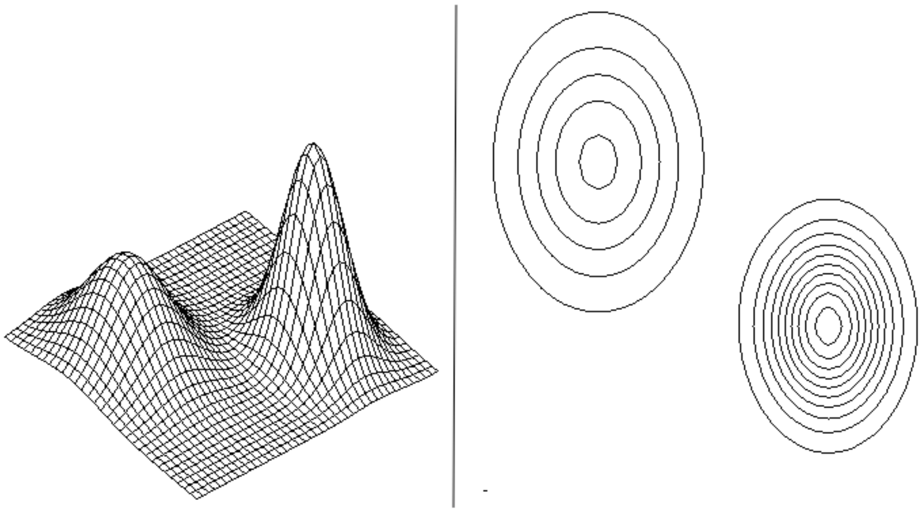


Figure 6.2: Multivariate two class classification problem

Univariate case with multiple classes can be seen in figure 6.3. There are three peaks matching three classes, and again the class with highest posterior probability is chosen. However, one correlation value does not carry enough information to be able to distinguish between three or more classes, so this case is not feasible. Also, there can be significant overlapping in the conditional probability distributions, which can render this method too unreliable for practical use.

Figure 6.4 shows a classification problem for two variables and three classes. The left side of the figure shows the multinormal distribution values for each class, and the right side shows the contours for each class. The peaks appear to be separate, but as can be seen in the contour plot, two of the classes are close enough to each other to cause some ambiguity. The simplest rule for choosing the class is the same

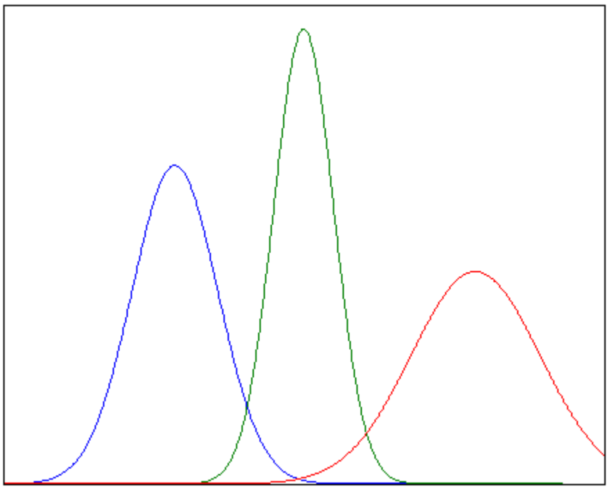


Figure 6.3: Multiclass univariate classification problem

as in the bivariate case, i.e., $\underset{k}{\arg\max}\left\{P\left(\omega_k\left|\mathbf{x}\right.\right)\right\}$. The cases with more than two variables cannot be visualized, but the principle stays the same: calculate posterior probabilities for the each sample vector $\mathbf{x_i}$ and then assign each to the class with highest probability.

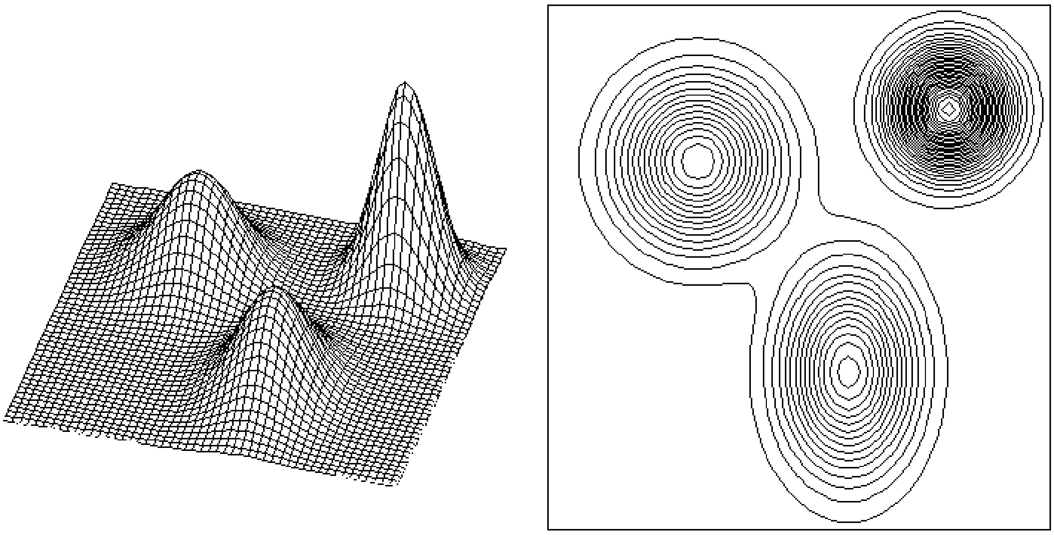


Figure 6.4: Two-variate normal distribution for three classes, and contour lines

Bayesian classification is an effective method for recognizing image sources, unless the distribution parameters for the classes are too close, i.e., if assigning a sample vector to a class is hard, since the posterior probabilities are too close.

6.4 Linear regression

The idea of linear regression in machine learning is to estimate coefficients from the training data. These coefficients can be used to compute scores of the dependent variable or variables, i.e., scalar y or vector $\mathbf{y} = [y_1, y_2, \ldots, y_k]^T$. In both cases there is a constant β_0 included in the equations, which is trained as well as the proper coefficients. Univariate model aims to either classify the images based on whether the image is from selected camera or not by calculating regression scores from the correlation between the chosen camera and the noise residuals, i.e., decide to which of the two classes the image belongs to. The problem can also be formulated as as multiclass one, i.e., from which camera the images are, but the correlations with just one camera makes this formulation unfeasible, since there is not enough data to make the decision. Therefore, only the first formulation is discussed here.

Univariate regression model is generated from correlation coefficients between the reference pattern of a specific camera C_k and the noise residuals of the training set images. The images are divided into two sets - training set A of images from camera C_k, and training set B of images from other cameras. The correlations are then computed between the camera reference pattern and the noise residuals from both sets to form matrix $\mathbf{X}$, augmented with ones on the left side to account for the constant β_0. Horizontal line separates sets A and B.

$$\mathbf{X} = \left[\begin{array}{cc} 1 & \rho_{A_1}^{C_k} \\ 1 & \rho_{A_2}^{C_k} \\ \vdots & \vdots \\ 1 & \rho_{|A|}^{C_k} \\ \hline 1 & \rho_{B_1}^{C_k} \\ 1 & \rho_{B_2}^{C_k} \\ \vdots & \vdots \\ 1 & \rho_{|B|} \end{array}\right]$$

Matrix $\mathbf{Y}$ contains the correct classes, i.e., ones for images from set A, and zeroes for image from set B.

$$\mathbf{Y} = \begin{bmatrix} \underbrace{1 \quad 1 \quad \ldots \quad 1}_{\text{Selected camera}} & \underbrace{0 \quad 0 \quad \ldots \quad 0}_{\text{Other cameras}} \end{bmatrix}^T$$

Since estimating linear regression coefficients, and naturally the constant β_0, is a least square error problem, the coefficients can be calculated as (Alpaydin, 2010)

$$\hat{\boldsymbol{\beta}} = \left(\mathbf{X}^T\mathbf{X}\right)^{-1}\mathbf{X}^T\mathbf{Y}$$

The univariate regression model is then applied to a test image by first calculating the correlation coefficient between the filtered image and the reference pattern of camera C_k, and then calculating the weighted sum. Score r_j of image I_j is then (Alpaydin, 2010)

$$r_j = \beta_0 + \beta_1\rho_j^{C_k}$$

Naturally, there is one more step in order to interpret score r, namely estimating the minimum regression score for assigning the image I to one of the classes. The easiest way to do this is to naïvely classify the training images with the regression model, and then find the minimum score for the images from the specific camera C_k, or the maximum score for images that are *not* from camera C_k. However, in case the classes overlap, this approach produces a lot of ambiguities for classification. It is also possible to use e.g. bayesian classification and assume that the scores are normally distributed. This approach, however, is beyond the scope of this assignment.

The problem can be formulated as multivariate linear regression, either with two classes, or multiple ones. The idea is to estimate the coefficients $\beta_0, \beta_1, \ldots, \beta_N$, where N is the number of cameras, and then calculate the score $r_i = \beta_0 + \sum_{j=1}^{N} \beta_j \rho_i^{C_j}$, i.e., calculate the weighted sum of correlations of noise residuals of image i with each of the cameras. If the problem is formulated as a one with two classes, the vector Y of the correct classes contains again ones for images from the specific camera C_k, and zeroes for the images not from that camera.

It is also possible to formulate the problem as multiclass classification problem. Then the vector Y contains the number of the correct class for the cameras, i.e., one for camera C_1, two for camera C_2 etc. The problem with multiclass formulation is that it quite probably contains overlapping classes, and needs a large training data set accounting for all the classes. Same goes for multivariate formulation, as it needs to provide sufficient amount of images to provide enough correlations for each of the variables.

Estimation of the coefficients is similar to the case of univariate classification. First a matrix $\mathbf{X}$ of correlation coefficients is formed and augmented with a vector of ones to the left side to account for the constant term β_0. The number N_i is the number of the images in all the training sets.

$$\mathbf{X} = \begin{bmatrix} 1 & \rho_1^{C_1} & \rho_1^{C_2} & \cdots & \rho_3^{C_N} \\ 1 & \rho_1^{C_1} & \rho_1^{C_2} & \cdots & \rho_3^{C_N} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \rho_{N_i}^{C_1} & \rho_{N_i}^{C_2} & \cdots & \rho_{N_i}^{C_N} \end{bmatrix}$$

Generation of the correct class vector $\mathbf{Y}$ is the key to selecting either the twoclass or multiclass model. If $\mathbf{Y}$ is formulated as ones for selected camera C_k, and zeroes for other cameras, the model will be similar to the univariate twoclass problem except for the number of coefficients β.

$$\mathbf{Y} = \begin{bmatrix} \underbrace{1 \;\; 1 \;\; \ldots \;\; 1}_{\text{Selected camera}} & \underbrace{0 \;\; 0 \;\; \ldots \;\; 0}_{\text{Other cameras}} \end{bmatrix}^T$$

Instead, if $\mathbf{Y}$ contains the number of the i^{th} camera specifying from which camera the training image came from, the problem is a multiclass one. $\mathbf{Y}$ is then

$$\mathbf{Y} = \begin{bmatrix} \underbrace{1 \;\; 1 \;\; \ldots \;\; 1}_{\text{Camera 1}} & \underbrace{2 \;\; 2 \;\; \ldots \;\; 2}_{\text{Camera 2}} & \ldots & \underbrace{N \;\; N \;\; \ldots \;\; N}_{\text{Camera N}} \end{bmatrix}^T$$

As with the univariate problem, estimating the coefficients is a least square error problem, for which the solution is (Alpaydin, 2010)

$$\hat{\boldsymbol{\beta}} = \left(\mathbf{X}^T X\right)^{-1} \mathbf{X}^T \mathbf{Y}$$

Computing the linear regression scores is analogous to the univariate case, except that there are more coefficients. Score r_j of image I_j is calculated as a weighted sum

Table 6.2: Mathematical models for linear regression

	Univariate Correlation of noise residuals with one camera	**Multivariate** Correlation of noise residuals with all cameras
Two classes	$r_j = \beta_0 + \beta_1 \rho_j^{C_k}$ $\mathbf{Y} = [1\,1\,\ldots\,1\,0\,0\,\ldots\,0]^T$	$r_j = \beta_0 + \sum_{k=1}^{N} \rho_j^{C_k}$ $\mathbf{Y} = [1\,1\,\ldots\,1\,0\,0\,\ldots\,0]^T$
Multiple classes	$r_j = \beta_0 + \beta_1 \rho_j^{C_k}$ $\mathbf{Y} = [1\ldots 1\,2\ldots 2\ldots N\ldots N]^T$	$r_j = \beta_0 + \sum_{k=1}^{N} \rho_j^{C_k}$ $\mathbf{Y} = [1\ldots 1\,2\ldots 2\ldots N\ldots N]^T$

of correlation coefficients between the image I_j and the camera C_i, and the constant β_0 is added.

$$r_j = \beta_0 + \sum_{i=1}^{N} \beta_i \rho_j^{C_i}$$

Estimation of the class boundaries is a huge problem with multiclass formulations, since there is bound to be a great amount of ambiguities. There are quite a few methods, e.g. stochastic properties of intra-class distributions. However, these are beyond the scope of this work. Also, the need to use estimation techniques for class boundaries requires, again, more data. As always, there is a tradeoff between accuracy and amount of training data.

Table 6.2 presents the mathematical models for linear regression. In the univariate both the univariate cases the model is created from the correlations between the noise residuals and only one camera. In the multivariate case the models is formed by taking into account the correlation of noise residuals with all the cameras. The number of variables decides, of course, the formulation of regression score r_j.

The number of classes, either two or more, determines how the result vector $\mathbf{Y}$ is formulated. Two-class formulation is that there are only ones for images from selected cameras, and zeroes for other images. If instead the vector contains the index of the camera the image was taken with, the case is a multiclass one. However it should be noted that the univariate multiclass case is not a feasible nor possible approach to the classification problem, since the differences of correlations between the images with the noise patterns from cameras it was not taken with are not large enough to allow for discrimination.

The choice between univariate and multivariate regression is not a simple one, the unfeasible univariate multiclass case notwithstanding. Multivariate regression considers the relationships between the correlations with all the reference pattern, whereas univariate regression focuses only on the correlation between the reference pattern of the selected camera and the noise residual of the image. The simplest possible formulation, i.e., univariate two-class regression, is useful and efficient when the task is to find which images are from the chosen cameras. Unfortunately, if all the images have to be classified, a lot of computing effort goes to waste: first, the images from camera 1 are classified and removed, then the images from camera 2, and so on. The worst case is $O(mn)$ comparisons, where m is the number of images and n is the number of cameras. Since an image contains a huge amount of information, all the images cannot be stored in non-permanent memory, and therefore one comparison is a computationally expensive operation.

Multivariate regression is more efficient method to classify a large amount of images, since each image has to be loaded only once and a reasonable amount of reference patterns can be kept in memory. However, training multivariate regression requires more data. Using multivariate regression to recognize which images are from selected camera is quite probably an overkill

It is possible to formulate the problem as non-linear and multinomial regression problem. Nonlinear univariate formulation can be reduced to linear regression. Nonlinear multivariate regression is beyond the scope of this work, due to the added complexity. The classification models should be kept as simple as possible, and therefore it is feasible to use the simplest efficient methods possible.

6.5 Tuning the source camera identification algorithm

The most serious bottleneck of using PNU patterns is the memory consumption. The image size is usually about 4000x3000 and there are three color layers, which gives the minimum boundary of 34 MB per image when the image is represented with 8 bit unsigned integers. Furthermore, the image will be converted into 32 bit double precision numbers, which gives the final estimate of 137 MB per image. Therefore it is crucial to find ways for reducing the memory consumption as much as possible.

One of the methods is using a cropped image, for example taking only 1024x1024 area from the image center. This also allows for reducing the boundary effects caused by filtering, as the image can be first cropped to size of about 1040x1040, filtering performed and then the final 1024x1024 is taken.

The structure of the Bayer pattern of color filter array (CFA) was explained in chapter 2. Image 2.2 showed how about half of the filters in the Bayer CFA were for green light. Therefore, it is possible to process only the green layer of the image, since it contains about half the information of the image signal.

When these methods are combined, the memory consumption per image is reduced to about 4 MB per image. Since in practice memory consumption correlates with amount of virtual memory swapping, this is a significant improvement. However, the trade-off is the need to use more effective filtering methods (e.g. wavelets or n$^{\text{th}}$ order differential filter) in order to compensate the diminished amount of data.

One cause of problems for the classification is the existence of more than one camera from the same camera model. Due to the similar manufacturing process, these cameras are not as easy to identify as the cameras of different models. It could be possible e.g. to first identify the camera models, and then use the identification process to discriminate between the individual cameras inside the groups of the camera models, i.e., a two-tiered identification process. This thesis contains an experimentation for evaluating this hierarchical model.

Chapter 7

Focusing topic

Starting point was on the one hand turning the noise caused by cameras into advantage and making it a fingerprint for forensics. On the other hand there was interest to applying image classification methods to this type of practical learning recognition problem. Promise of matching these appeared to be in relying on what image data already contains (not adding some tags) and that problem and method were about rough recognition of possible target for further forensics (not exact identification and proof). The aim was to discover whether a promising research direction can be found for further research.

This chapter describes how the topic is focused. First, the specifics of the source data and the tasks area give in 7.1. The utilization of the image source database is discussed in chapter 7.2. Chapter 7.3 enumerates the decisions that have been made in order to carry out the experimentations. Finally, chapter 7.4 introduces the hypotheses to be validated or refuted by exploiting the experimentation results.

7.1 Topic and focusing

The task is to find out whether images can be attributed to the camera they were taken with, and how well it can be done. There is a known set of cameras and for every image it is known with which camera the image was taken with.

An existing image base, i.e., Dresden forensic image database (Gloe and Böhme, 2010) will be used in the experimentations. The database will be discussed more later in chapter 7.2.

It was decided that a set of 24 cameras will used in the thesis. The manufacturers, models and number of cameras per each model are shown in table 7.1. All the cameras are not used in all the experimentations, but a subset is selected according to the needs of the experimentation. The number of cameras, 24, is large enough to show reliably whether the method works, but small enough to keep the experimentations manageable. Choosing cameras so that there are more than one camera from most models gives a possibility to examine both inter-model and intra-model differences of cameras.

Manufacturer	Model	#
Canon	Powershot A 640	1
Olympus	mju-1050sw	5
Panasonic	DMC-FZ50	3
Pentax	Optio W60	1
Ricoh	GX100	5
Samsung	NV15	3
Sony	DSC-T77	4
	DSC-W170	2
Total		24

Table 7.1: Selected camera models

7.2 Utilizing image source

Dresden forensic image database (Gloe and Böhme, 2010) is used as the source of images. The database was created in order to have a standardized benchmarking image sets for developing algorithms to be used in digital forensics. The image acquisition and storage procedures have been carefully designed to minimize the effect of post processing and lossy compression. The resolution of the images is high, over 2800 pixel wide and over 2100 pixels high. Figure 7.1 shows some sample images from the Dresden image database.

Figure 7.1: Sample images from the Dresden forensic image database (Gloe and Böhme, 2010)

To gain enough data, 9014 images from 74 different cameras were selected. Of these, there were 27 different camera models from 12 different manufacturers. However, only some subsets of the source database will be used in the experimentations due to performance considerations. The subsets are selected so that the resolution of cameras in the subset are the same, in order to minimize the effect of image resizing algorithms to preserve the PNU fingerprints, and also to fulfill the condition of synchronization. Another reason besides the effect of resizing algorithms was that resizing is a computationally complex procedure with large images, and without a sufficient parallel computing facilities it would take the largest part of the processing time.

All the images of the selected 24 cameras were filtered before the experimentations with gaussian and wavelet filter and cropped to 1024x1024 region from the middle of the green layer, in order to avoid recomputing everything in every experimentation. Furthermore, the pixel non-uniformity noise (PNU) fingerprints were computed in the beginning.

A further advantage of choosing these camera models is that all the models have identical image sizes. This fulfills the condition of *synchronization* given in chapter 5, i.e., the images do not need scaling, cropping or rotation. This way, the differences between camera models and individual cameras are not overlapped by the differences in the spatial representation of the images. Possibilities to ignore the differences in the spatial representation are not a topic of thesis, but certainly one for the further research.

7.3 Source identification decisions

Twenty four cameras will be chosen for the experimentation, some of which are of the same model. The most important factor affecting this choice are the identical image sizes, which allows for speeding up the training and recognition process by eliminating the need for resizing the images. The image resizing is a time consuming process due to the extremely high quality and large size of the images in the database, and is not an interesting factor in this experimentation. Even though in this experimentation only 1024x1024 region from the green channel is to be used, the images would have to be resized before cropping process. As the algorithm is spatial in nature, without image resizing the patterns would be of different sizes, which would in turn have a serious effect on the recognition accuracy. Furthermore, smaller size allows for faster noise pattern filtering - some of the filtering methods, e.g. wavelet and nth order differential, are time consuming as such. It is possible to create noise residuals beforehand, but that does not speed up the process significantly, since loading and saving takes a lot of time, too.

Naïve bayesian classifier will be used in this experimentation, namely multivariate, multiclass one. The input data will be a vector of correlations between the noise pattern and the trained fingerprints, and the output the class for the noise pattern, and the probabilities for matching the noise pattern with the fingerprints. In this experimentation the images will all be from known cameras, i.e., there are no noise residuals that do not match any fingerprint. This allows for using simpler recognition algorithms, since it is not necessary to determine lower bounds for correlations, and it is not necessary to alternate the bayesian recognition algorithm to handle cases for which the a posteriori probabilities are very low and therefore almost equal.

Principal component analysis will not be used for preprocessing the correlations in this experimentation. Aim of the experimentation is to validate the idea of using bayesian classifier in conjunction with noise patterns and fingerprints, and therefore as pure data as possible is needed. Using principal component analysis and other classifiers is out of the scope of this experimentation.

The noise residuals in the inter-model classification experiment will be calculated using both gaussian and wavelet filtering, and 50 and 60 training images. In the subsequent experimentations only the filtering method with best performance is used. Same images will be used for creating the fingerprints and matching the noise residuals, which allows for intermediate storage of the noise residuals to avoid repeated cropping and filtering. Even though using the same noise residuals for creating the fingerprints and matching produces overly optimistic results, it is important to validate the underlying principle before carrying experimentations in more complicated situations.

7.4 Hypotheses

Defining the focus of the experimentations, choosing the source data and specifying the methods lead to two hypotheses:

1. Using multivariate multiclass bayesian classifier on the correlation vectors between image noise residual and image fingerprint provides reasonably accurate image source identification method.

2. The posterior probabilities for correct and incorrect classifications are from different kinds of distributions.

Hypothesis 1 is directly related to the usefulness and feasibility of the source identification method. If sufficient accuracy cannot be reached, testing further hypotheses is meaningful only for analysing the problems in the method. Reasonably accurate means here $> 95\%$ correct recognitions, as the method does not give indisputable proof, but to allow for choosing interesting images for further investigation.

Hypothesis 2 is related to the stability of the identification system. Posterior probability defines certainty of a decision, i.e., certain decisions have a high posterior probability and uncertain decisions a low posterior probability. If the correct decisions are certain and incorrect decisions uncertain, the decisions are less likely to be ambiguous or random.

Hypothesis 2 can be validated by examining the histograms of posterior probabilities in the case of correct and incorrect decisions. The criterion for the difference of distributions is that histograms are clearly different. Also, the posterior probabilities of correct classifications are significantly more certain, i.e., nearer to 1.0, than the posterior probabilities of incorrect classifications.

In the context of evaluating correctness of the classifications, the posterior probability is called “certainty”. For example, it could be said that“the system has classified the sample i correctly with certainty P”, compared with regular posterior probability stating that “the probability of sample i of being class ω_k is P”.

Chapter 8

Experimentation plan

The first experimentation introduces the basic case of image fingerprinting, i.e., recognizing different cameras. Eight camera models and one camera from each model were chosen for this experimentation. Since there are two parameters with two possible values, i.e., filtering method and number of the reference images, the experimentation is repeated four times. The results of this experimentation determine which parameters will be used in the rest of the experimentations. The number of images in total in this experimentation is 840. Since this is quite a large number and the overhead caused by loading and filtering of high resolution images is considerable, the images are prefiltered.

The second step will use the best parameters from the first experimentation. The aim is to recognize different cameras belonging to the same model; in total, 3 camera models will be used, containing 12 cameras in total.

The third experiment will determine whether it is possible to recognize different cameras with no information on the camera model. Again, the best parameters from the first experimentation are utilized. Twenty for cameras from eight different models are used, but all the cameras are put in the same pool. No a priori information on the camera models is used to support the classification. In total 1111 images will be used in this experimentation.

The fourth experimentation will determine whether it is possible to generate model-specific fingerprints, i.e. recognize all the cameras from the same model by using reference patterns from only one camera from the model. Five camera models will be used, resulting in total 20 cameras and 2179 images. Again, the parameters from the first experimentation will be used.

In total, the size of the image pool is considerable, over 2000 images, and therefore only appropriate but subsets of the images will be used. The subsets have been chosen so that they can be expected to represent the data set well enough.

8.1 Data collection

Three different types of data will be collected during the experimentation:

- Correlations between each noise pattern and each camera fingerprint
- Classification results, i.e., class and a posteriori probabilities for each class

- Performance metrics

Correlations are used as the input to the classification system. Classification results can be then used to evaluate the classification method. Finally, the performance metrics will be used to fine tune the method as well as to analyze the feasibility.

Correlations

Two-dimensional Pearson correlation coefficient will be calculated between each image and each camera. Table 8.1 shows an example of the resulting correlation matrix. Each row of the table refers to the image correlated with the cameras. Each column refers to the camera fingerprints, except the first column which contains the image number.

Table 8.1: Correlations between image noise patterns and camera fingerprints

Image	Camera 1	Camera 2	Camera 3	...	Camera M
1	0.1232	0.0232	0.0012	...	0.00012
2	0.0001	0.0033	0.1002	...	0.012
3	0.1502	0.0933	0.1740	...	0.0020
...	...	...	...	...	...
N	0.0099	0.029	0.0017	...	0.1750

Since a multivariate naïve Bayesian classifier is used, it is not possible to derive conclusions from just one correlation. The final classification depends on the relationships of the correlations between the noise pattern and its correlation with all the camera fingerprints. For example, row 3 in table 8.1 shows a situation where the correlations with camera 1 and camera 3 are nearly identical. However, the class is decided based on the other correlations – the correlations do not automatically imply ambiguity or that the class would be camera 3.

Classification results and probabilities

In order to derive conclusions on the classes, it is necessary to collect the classification results and the posterior probabilities for each camera. Table 8.2 shows collected data with the image number in the 1st column, class in the second column, and the posterior probabilities in the rest of the columns.

Image	Class	$P(C_1\|x_j)$	$P(C_2\|x_j)$	$P(C_3\|x_j)$	...	$P(C_M\|x_j)$
1	3	0.12	0.02	0.82	...	0.18
2	1	0.74	0.13	0.102	...	0.05
3	M	0.14	0.093	0.52	...	0.59
⋮	⋮	⋮	⋮	⋮	⋮	⋮
N	2	0.05	0.77	0.002	...	0.1750

Table 8.2: Classes and posterior probabilities

Usually the differences in the posterior probabilities are large enough to be unambiguous. However, row 3 shows a situation where the posterior probabilities of camera 3 and camera M are near to each other. Unfortunately this kind of ambiguity is not easy to handle - a bayesian classifier would still classify the image to class M, even though the decision is quite uncertain.

Performance metrics

In addition to the data pertaining to the image source recognition, also performance metrics are collected. These include the preprocessing time for each image, time needed for classification, and processor time, memory and virtual memory consumption. These metrics can be used to fine tune the algorithm, and to compare different filtering and cropping techniques. Also the accuracy of classification, and false acceptance and false rejection rates are recorded in order to validate the accuracy and reliability of the image source recognition algorithms. The data will be examined using visualizations, but no further hypotheses or conclusions will be drawn.

8.2 Analysis of classification

There are two important aspects in analyzing the performance of the classifier:

1. Inter-model classification, i.e., how accurately the classifier can make decision between camera models
2. Intra-model classification, i.e., how accurately the classifier can distinguish the different individuals from the same camera model

As stated before, it should be relatively easy to recognize the camera models due to the different manufacturing processes, even if the camera models are from the same manufacturer. However, since the individual cameras of the same model are created by the same process, it is plausible that their fingerprints are closely related. As for now, there is not sufficient metrics for actually defining what “closely related” means exactly. The only way is to examine the classification results and to decide whether they are “similar enough” - a subjective and error prone approach.

There are two alternatives for analyzing the recognition results:

1. using two-tier process, i.e., first recognizing the camera model by combining the images from individual cameras of the same model, and using them as a wholeness. After the camera model has been recognized, the recognition process is done again, but this time only the individual cameras from the camera model in question are used as the input for the recognition process
2. not making distinction between individual cameras and camera models, and hoping that the individuals are dissimilar enough to be accurately recognized, i.e., one-tiered process

Both of these approaches have their merits. If the purpose is to only sift through the images to find the potentially interesting cameras, the first option is quicker. However, if some kind of proof of image source is needed, then the second alternative is better. Both the alternatives are examined in this thesis, as they will give insight in the identification system whether or not the hypotheses are fulfilled.

In addition to these two alternatives, the existence of model specific fingerprints will be examined. These fingerprints would give a possibility to recognize a blind source camera, provided that there are some cameras of the same model among the known cameras. This kind of process would expand the usefulness of the identification process significantly.

Chapter 9

Experimentation results

There are four experimentations:

1. Inter-model: identification of individual cameras, one from each model, i.e., whether it is possible to discern between individual cameras when each is of different model.
2. Intra-model: identification of individual cameras when only one camera model is considered
3. Identification of individual cameras using naïve approach: all the cameras are recognized individually even when there are more than one camera from each model
4. Model specific fingerprints: an individual camera is used for testing and the rest cameras from the same model for creating a model specific fingerprint

Experimentations 1 and 2 are the two sides of proving whether it is even possible to use this method to classify cameras. Experimentation 1 shows that the variations in imperfections are large enough to allow for discerning between camera models. Experimentation 2 examines the variation inside one camera model to show that the variation is large enough to allow for classifying individual cameras. If the work would be carried out without this split, it is probable that the inter-model variations would hide the much smaller intra-model variations, and the problems would be hard to find. Experimentation 3 gives a glimpse to what happens if the distinction between inter-model and intra-model recognition is not done, i.e., the cameras are recognized individually even if there are more than one camera from each model. Experimentation 4 explores briefly the possibility of generating model specific fingerprints, i.e., identification of the model of an unknown camera based on the cameras from the training set.

All the cameras are assumed to have the same image size (3648x2736). This will prevent the image resizing algorithms from affecting the results or the repeatability of the experimentation. Also, experimenting with both filters (gaussian and wavelet) and sample sizes (50 or 60) were done in the first part, since that demonstrated the effectiveness of using wavelet filter with sample size 60.

9.1 Inter-model classification

In the inter-model classification experimentation there were eight different camera models, summarized in table 9.1. Only two of the models were from the same manufacturer, i.e., Sony. The other cameras are from different manufacturers. This experimentation considers whether there are large enough differences in PRNU fingerprints to allow for inter-model identification.

Class	**Manufacturer**	**Model**	**Images**
1	Canon	PowerShot A640	85
2	Olympus	mju-1050SW	109
3	Panasonic	DMC-FZ50	143
4	Pentax	Optio W60	91
5	Ricoh	GX100	93
6	Samsung	NV15	130
7	Sony	DSC-T77	90
8	Sony	DSC-W170	99
Total			**840**

Table 9.1: Camera models for inter-model identification

The data set is further divided into reference images and test images. For each camera, $N_{ref} \in \{50, 60\}$, and therefore $N_{test} = N - N_{ref}$. For example, the first camera, Canon PowerShot A640, has 85 images in total, and therefore $N_{test} = 35$, if N_{ref}=50, and N_{test}=25, if $N_{ref} = 60$.

Lukas et al. suggest (Lukáš et al., 2006) that number of reference images $N_{ref} > 50$. Thus, to establish the baseline for identification, reference image sample sizes 50 and 60 were used to construct the camera fingerprints. Flat fielding was not an option, since it was not possible to gain access to the original cameras.

There were two methods to filter the images for PRNU fingerprinting, namely gaussian and wavelet filtering. Therefore, in total four different passes were needed for the experimentation: gaussian filtering with 50 and 60 reference images, and wavelet filtering with 50 and 60 reference images.

Correct classifications

Figure 9.1 shows the ratios of correctly classified images for every combination of filter and number of reference images. The blue bar and cyan bars are for gaussian filtering with 50 and 60 reference images, respectively. The yellow and red bars for wavelet filtering with 50 and 60 reference images, respectively. The percentages are collected to table 9.2.

As can be seen in both figure 9.1 and table 9.2, decent results can be achieved with all filtering techniques, except with camera 6. Gaussian filtering classifies correctly 86.1% of the images, with lowest ratio 57.1% for camera 6. However, the configuration with wavelet filter and $N_{ref} = 60$ is clearly the best: all ratios are $\geq 90.0\%$. Camera 7 exhibits an anomaly, since the recognition results are somewhat lower for gaussian $N_{ref} = 60$ than for gaussian $N_{ref} = 50$. The reason is not clear, but since the difference is not a large one, the reason might be the random division

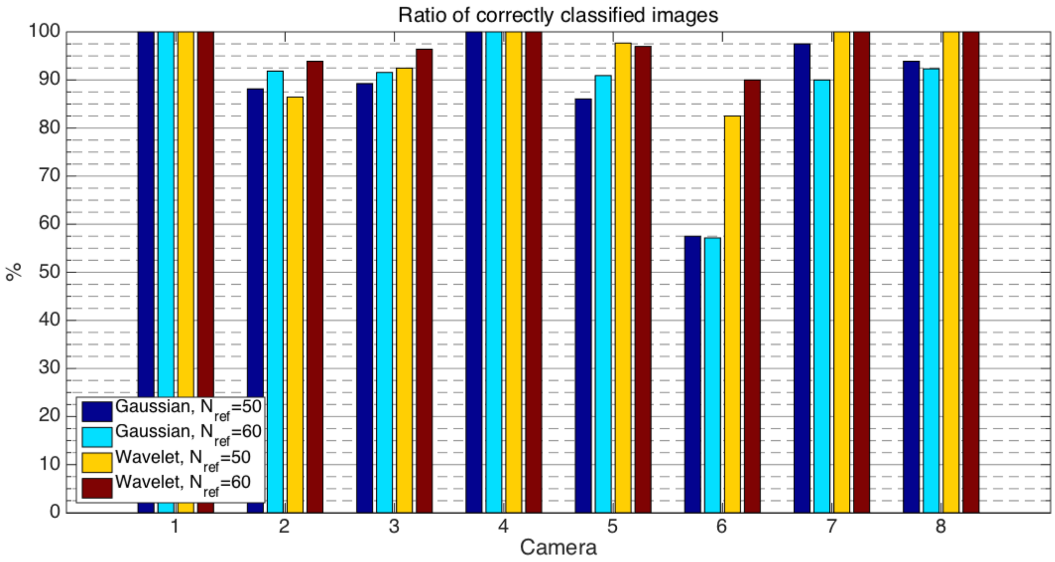


Figure 9.1: Inter-model classification: ratios of correctly classified images

of images to reference and test set, i.e., the difference is due to variation in data, not due to the method.

	Gaussian		**Wavelet**	
Camera	50	60	50	60
1	100%	100%	100%	100%
2	88.1%	91.8%	86.4%	93.9%
3	89.2%	91.6%	92.5%	96.4%
4	100%	100%	100%	100%
5	86.0%	90.9%	97.7%	97.0%
6	57.5%	57.1%	82.5%	90.0%
7	97.5%	90.0%	100%	100%
8	93.9%	92.3%	100%	100%
Total	**86.1%**	**86.1%**	**93.2%**	**96.1%**

Table 9.2: Ratios of correctly classified images

Relationship of target and predicted classes

Figure 9.2 shows the number of target and predicted classes for each camera, filter and N_{ref} combination. Blue bars represent the target frequencies, i.e., how many images there really are for each camera. Yellow bars represent the predicted frequencies, i.e., how many images are attributed to the camera.

In the figure 9.2, the most notable phenomenon is that the number of predicted images for each camera is higher than the real value, except for cameras 3 and 6. For the latter, some of their images are attributed to other cameras. The differences between the frequencies of target and predicted classes naturally decrease when the accuracy of the method increases. However, for cameras 3 and 6, even using the

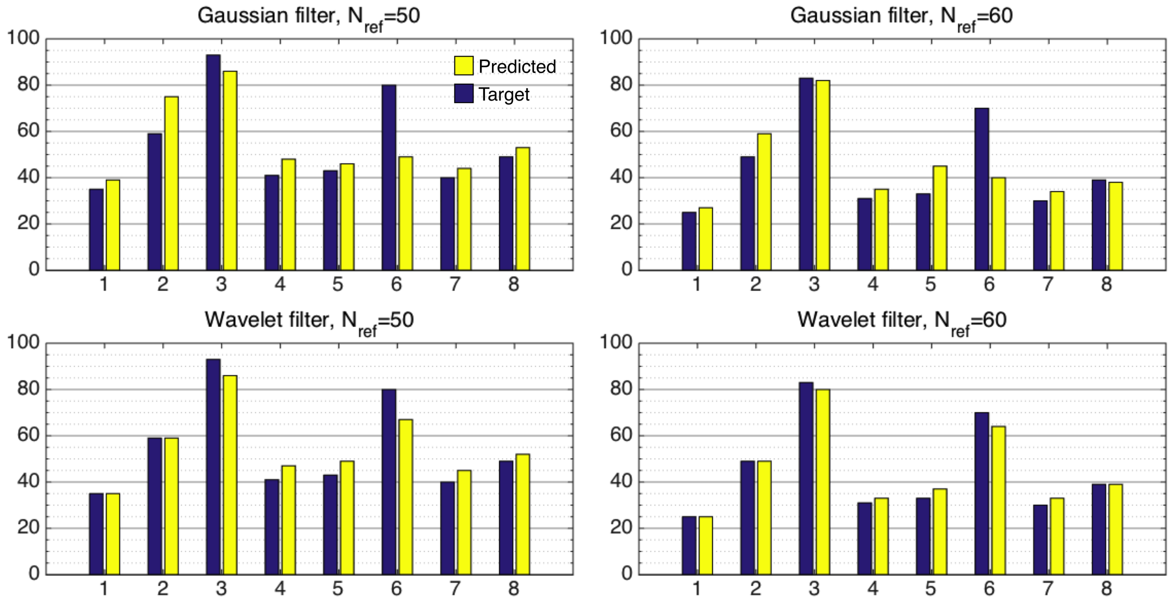


Figure 9.2: Inter-model classification: Frequencies of target and predicted classes

best method does not completely remove this problem: still some of the images from cameras 3 and 6 will be attributed to other classes.

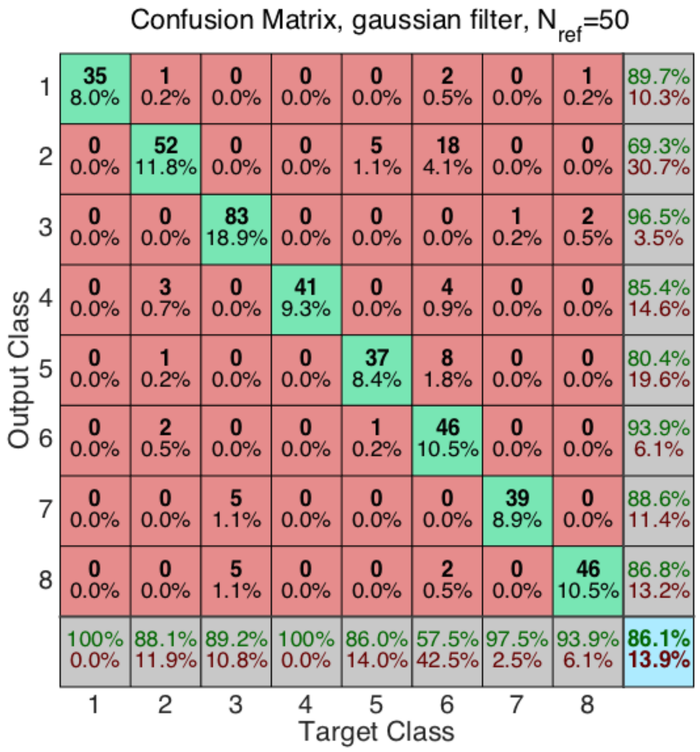
Confusion matrices

The confusion matrices for the inter-model classification experimentation are in figures 9.3 - 9.6. The columns represent the target classes, and rows the output, or predicted, classes. For example, in figure 9.3 the cell in row 5 column 6, later referred as cell (5,6) shows that 8 images from camera 6, i.e. Samsung NV15 in table 9.1 are classified as coming from camera 5, i.e., Ricoh GX100. Vice versa, cell (5,6) shows also, that 8 images that were classified to coming from class 5, i.e., Ricoh GX100, actually came from camera 6, i.e., Samsung NV15. As can be seen, Ricoh GX100 was the hardest camera to identify, and most often mistaken for camera 2, i.e., Olympus mju–1050SW. Cameras 1, 4, 7 and 8 were almost always identified correctly.

Most wrong attributions were made with camera 2 and 5, when gaussian filtering was used. Moreover, using 60 reference images with gaussian filtering actually caused more wrong attributions to camera 5 than using 50 reference images. The cause of this phenomenon is unknown. With wavelet filtering the attributions to camera 5 were most often wrong, the accuracy improving as it was expected to, with using 60 reference images instead of 50.

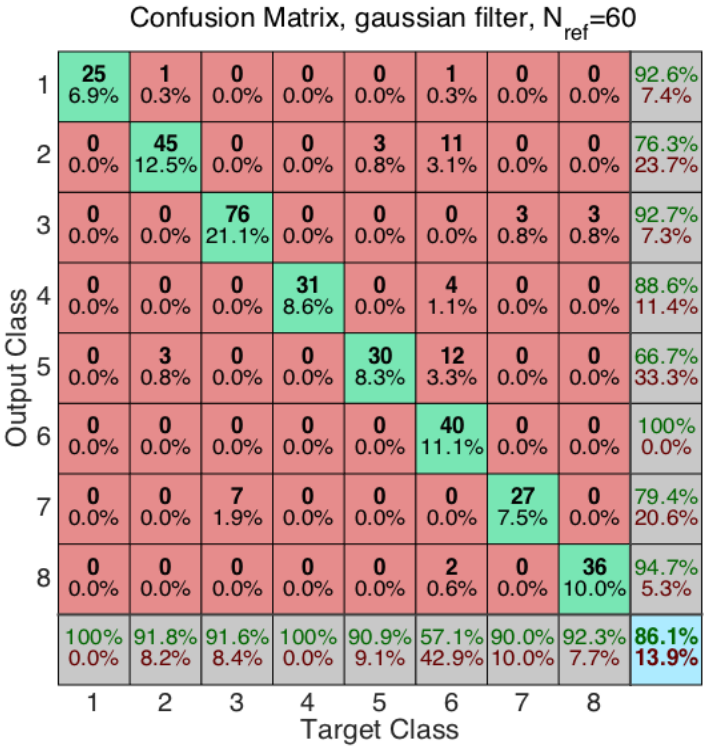
Anatomy of misclassifications

In order to understand how the system could be enhanced, it is important to examine the misclassifications to understand their structure and reasons behind them. It is not a straightforward process, but can give insight into the sources of errors.


Confusion Matrix, gaussian filter, N_ref=50
Output Class
Target Class

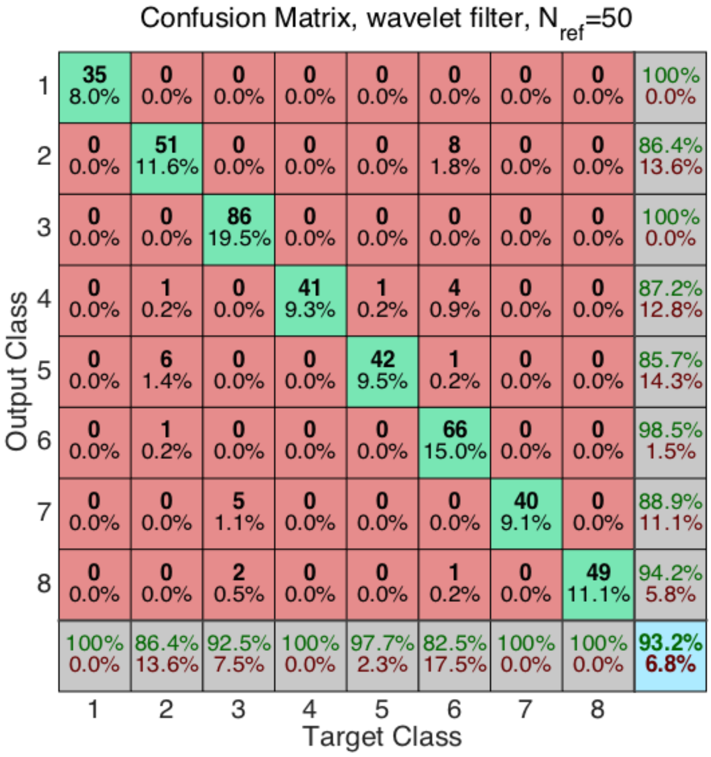
	1	2	3	4	5	6	7	8	
1	35 8.0%	1 0.2%	0 0.0%	0 0.0%	0 0.0%	2 0.5%	0 0.0%	1 0.2%	89.7% 10.3%
2	0 0.0%	52 11.8%	0 0.0%	0 0.0%	5 1.1%	18 4.1%	0 0.0%	0 0.0%	69.3% 30.7%
3	0 0.0%	0 0.0%	83 18.9%	0 0.0%	0 0.0%	0 0.0%	1 0.2%	2 0.5%	96.5% 3.5%
4	0 0.0%	3 0.7%	0 0.0%	41 9.3%	0 0.0%	4 0.9%	0 0.0%	0 0.0%	85.4% 14.6%
5	0 0.0%	1 0.2%	0 0.0%	0 0.0%	37 8.4%	8 1.8%	0 0.0%	0 0.0%	80.4% 19.6%
6	0 0.0%	2 0.5%	0 0.0%	0 0.0%	1 0.2%	46 10.5%	0 0.0%	0 0.0%	93.9% 6.1%
7	0 0.0%	0 0.0%	5 1.1%	0 0.0%	0 0.0%	0 0.0%	39 8.9%	0 0.0%	88.6% 11.4%
8	0 0.0%	0 0.0%	5 1.1%	0 0.0%	0 0.0%	2 0.5%	0 0.0%	46 10.5%	86.8% 13.2%
	100% 0.0%	88.1% 11.9%	89.2% 10.8%	100% 0.0%	86.0% 14.0%	57.5% 42.5%	97.5% 2.5%	93.9% 6.1%	**86.1%** **13.9%**

Figure 9.3: Inter-model classification: confusion matrix for gaussian filter, $N_{ref} = 50$


Confusion Matrix, gaussian filter, N_ref=60
Output Class
Target Class

	1	2	3	4	5	6	7	8	
1	25 6.9%	1 0.3%	0 0.0%	0 0.0%	0 0.0%	1 0.3%	0 0.0%	0 0.0%	92.6% 7.4%
2	0 0.0%	45 12.5%	0 0.0%	0 0.0%	3 0.8%	11 3.1%	0 0.0%	0 0.0%	76.3% 23.7%
3	0 0.0%	0 0.0%	76 21.1%	0 0.0%	0 0.0%	0 0.0%	3 0.8%	3 0.8%	92.7% 7.3%
4	0 0.0%	0 0.0%	0 0.0%	31 8.6%	0 0.0%	4 1.1%	0 0.0%	0 0.0%	88.6% 11.4%
5	0 0.0%	3 0.8%	0 0.0%	0 0.0%	30 8.3%	12 3.3%	0 0.0%	0 0.0%	66.7% 33.3%
6	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	40 11.1%	0 0.0%	0 0.0%	100% 0.0%
7	0 0.0%	0 0.0%	7 1.9%	0 0.0%	0 0.0%	0 0.0%	27 7.5%	0 0.0%	79.4% 20.6%
8	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	2 0.6%	0 0.0%	36 10.0%	94.7% 5.3%
	100% 0.0%	91.8% 8.2%	91.6% 8.4%	100% 0.0%	90.9% 9.1%	57.1% 42.9%	90.0% 10.0%	92.3% 7.7%	**86.1%** **13.9%**

Figure 9.4: Inter-model classification: confusion matrix for gaussian filter, $N_{ref} = 60$



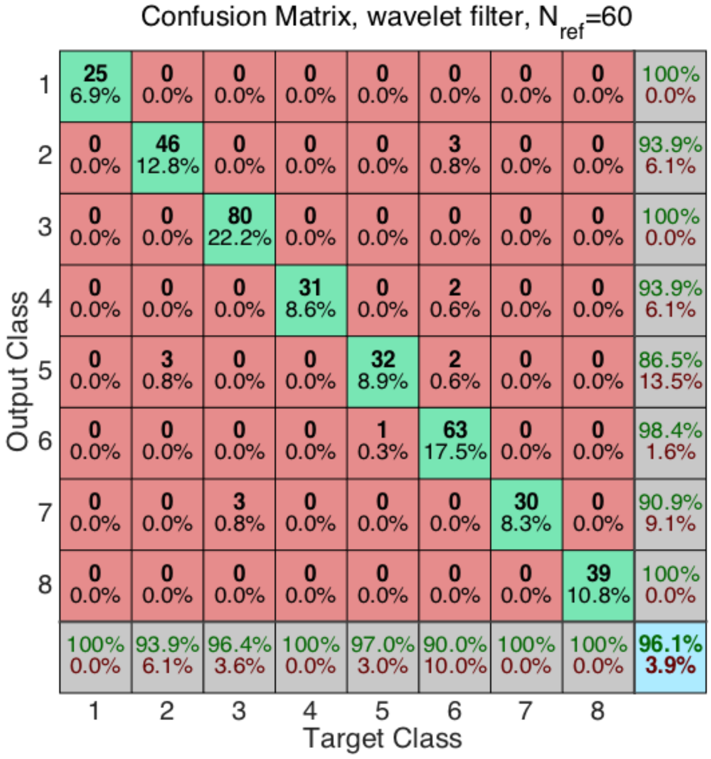
Confusion Matrix, wavelet filter, N_{ref}=50

Output Class

	1	2	3	4	5	6	7	8	
1	35 8.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
2	0 0.0%	51 11.6%	0 0.0%	0 0.0%	0 0.0%	8 1.8%	0 0.0%	0 0.0%	86.4% 13.6%
3	0 0.0%	0 0.0%	86 19.5%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
4	0 0.0%	1 0.2%	0 0.0%	41 9.3%	1 0.2%	4 0.9%	0 0.0%	0 0.0%	87.2% 12.8%
5	0 0.0%	6 1.4%	0 0.0%	0 0.0%	42 9.5%	1 0.2%	0 0.0%	0 0.0%	85.7% 14.3%
6	0 0.0%	1 0.2%	0 0.0%	0 0.0%	0 0.0%	66 15.0%	0 0.0%	0 0.0%	98.5% 1.5%
7	0 0.0%	0 0.0%	5 1.1%	0 0.0%	0 0.0%	0 0.0%	40 9.1%	0 0.0%	88.9% 11.1%
8	0 0.0%	0 0.0%	2 0.5%	0 0.0%	0 0.0%	1 0.2%	0 0.0%	49 11.1%	94.2% 5.8%
	100% 0.0%	86.4% 13.6%	92.5% 7.5%	100% 0.0%	97.7% 2.3%	82.5% 17.5%	100% 0.0%	100% 0.0%	**93.2%** **6.8%**

Target Class

Figure 9.5: Inter-model classification: confusion matrix for wavelet filter, $N_{ref} = 50$



Confusion Matrix, wavelet filter, N_{ref}=60

Output Class

	1	2	3	4	5	6	7	8	
1	25 6.9%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
2	0 0.0%	46 12.8%	0 0.0%	0 0.0%	0 0.0%	3 0.8%	0 0.0%	0 0.0%	93.9% 6.1%
3	0 0.0%	0 0.0%	80 22.2%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
4	0 0.0%	0 0.0%	0 0.0%	31 8.6%	0 0.0%	2 0.6%	0 0.0%	0 0.0%	93.9% 6.1%
5	0 0.0%	3 0.8%	0 0.0%	0 0.0%	32 8.9%	2 0.6%	0 0.0%	0 0.0%	86.5% 13.5%
6	0 0.0%	0 0.0%	0 0.0%	0 0.0%	1 0.3%	63 17.5%	0 0.0%	0 0.0%	98.4% 1.6%
7	0 0.0%	0 0.0%	3 0.8%	0 0.0%	0 0.0%	0 0.0%	30 8.3%	0 0.0%	90.9% 9.1%
8	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	39 10.8%	100% 0.0%
	100% 0.0%	93.9% 6.1%	96.4% 3.6%	100% 0.0%	97.0% 3.0%	90.0% 10.0%	100% 0.0%	100% 0.0%	**96.1%** **3.9%**

Target Class

Figure 9.6: Inter-model classification: confusion matrix for wavelet filter, $N_{ref} = 60$

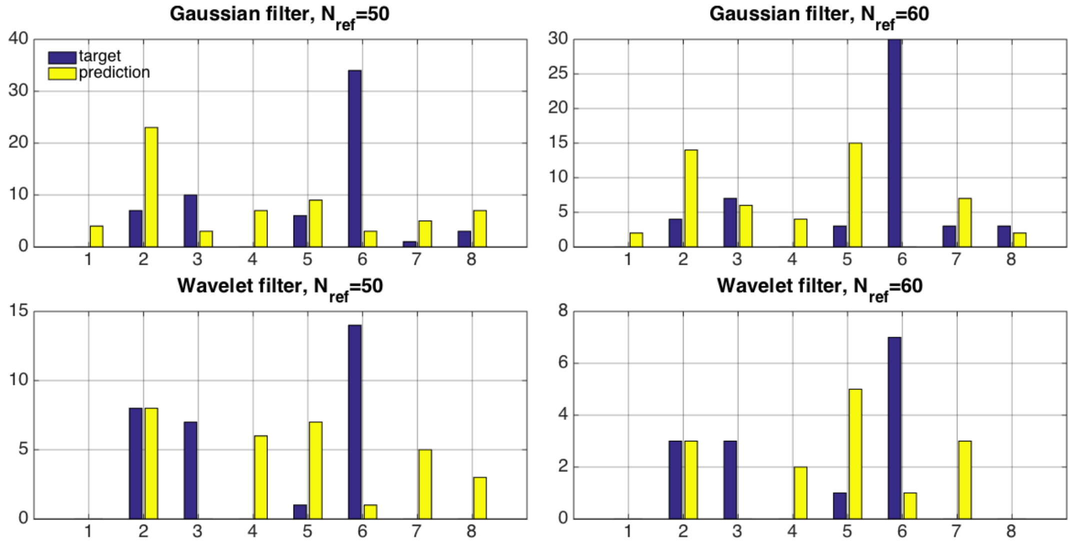


Figure 9.7: Inter-model classification: Target and predicted classes of misclassifications

Figure 9.7 shows the behavior of wrong predictions. Yellow bars represent the predicted classes and blue bars the target classes. In effect, this means that yellow bars show that images from other cameras have been attributed to that camera. Blue bars show that images from the camera in question have not been attributed to that camera. For example, all images from camera 1 are classified correctly (no blue bar), but in addition some images from other cameras have been assigned to camera 1 (yellow bar). As can be seen, there is a high spike on camera 6: many images from that camera have been assigned to other cameras. This can be seen as high yellow bars on other cameras, especially camera 2; extra images attributed to other cameras have to come from somewhere, in this case from camera 6.

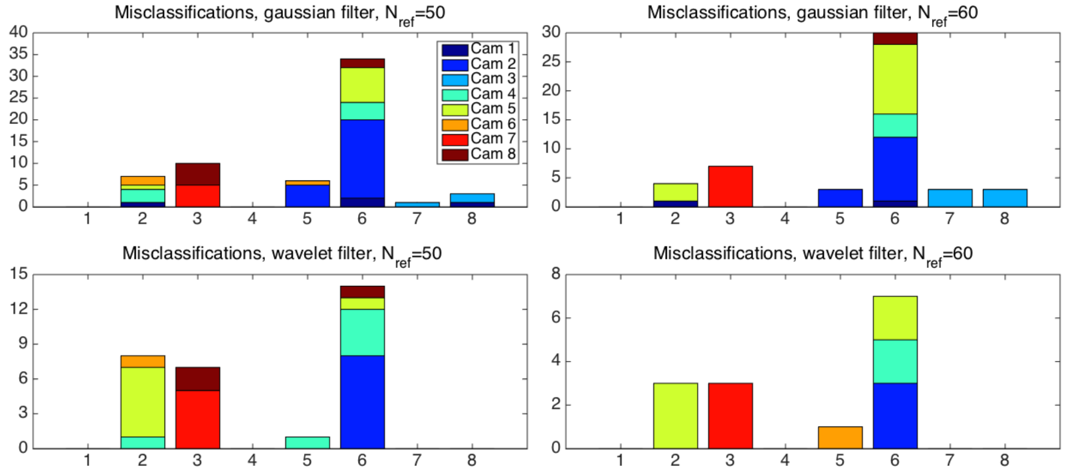


Figure 9.8: Inter-model classification: structure of misclassifications

Figure 9.8 shows the structure of misclassifications. The images from cameras 1 and 4 are always classified correctly. Camera 6 has the most misclassifications: the images from camera 6 are most often attributed to camera 2, with camera 5 being the second alternative. It should be noted that the y axis scales vary. Attribution of an image to camera 5 happens also for camera 2. And attribution to camera 2 for images from camera 5 exist, even though they are rare. This data implies strongly that for some reason, the fingerprints from camera 2 and 5 are easily confused with each other. The reason is left for further study.

It can be seen that figure 9.7 and figure 9.8 point to same direction: when the structure of misclassifications for camera 6 was examined, it explained the high yellow bar spikes on camera 2 in figure 9.8, i.e., the extra images attributed to camera 2 come from camera 6, and there are many images from camera 6 that are attributed to camera 2, and in some cases, camera 5.

One more thing to note in figure 9.8: the classification errors are not uniformly distributed or random. Misclassifications for each camera tend to fall into only some classes, not all, especially when the accuracy of the method increases. In practice this means that the classification errors for e.g. camera 2 tend to be attributions to camera 5, and not to others, and therefore there could be a method to further remove errors from the process

Certainty of classifications

Final thing to examine is the certainty of classifications, both for correct and wrong ones. The certainty can be examined as shown as a histogram in figure 9.9: x axis corresponds to posterior probability categories, and y axis shows the ratio of the posterior probabilities falling into that category.

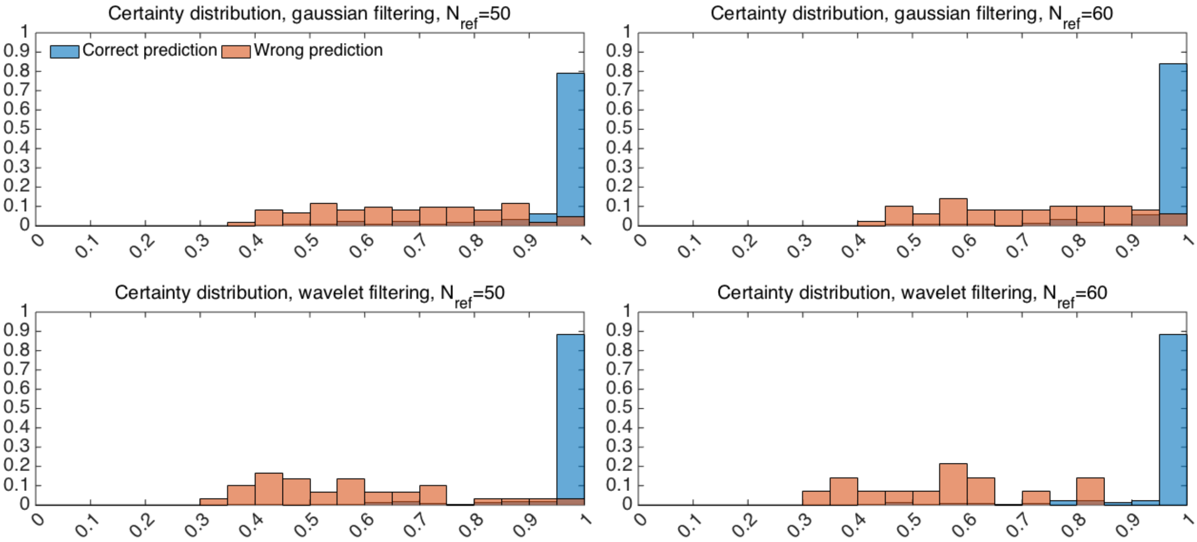


Figure 9.9: Inter-model classification: posterior probabilities for correct and wrong predictions

As can be seen in figure 9.9, the posterior probability, or certainty, is always very high for the correct classifications. This is a good sign: when the system makes a positive and correct identification, it is always "certain" of its correctness.

However, examining the histogram of wrong predictions shows that the classifications are fairly uncertain. This can give a rise for futher development of the system: it is possible to examine the posterior probabilities and accept the decision if the system is certain of it, and submit it to further examination if the decision is uncertain. Uncertainty means that the system has only bad choices and it has to choose one based on very slight differences.

Evaluation of hypotheses

Hypothesis 1 is used to estimate the feasibility of the method. As can be seen in figures 9.3 - 9.6, only the wavelet filtering with 60 reference images fulfills the criterion of accuracy being 96.1% $\geq$ 95%, in hypothesis 1. However, the wavelet filtering with 50 reference images comes close with accuracy 93.2%. Therefore wavelet filtering with 60 reference images is selected for later experimentations. The accuracies can be seen in the bottom right cell of the confusion matrices.

Figure 9.9 shows the histograms of relative frequencies of posterior probabilities, blue color marking the correct predictions, and orange color the incorrect predictions. As can be seen, the certainty of correct predictions is almost always very near to 1, i.e., very near to full certainty, with $\mu = 0.9698$ and $\sigma = 0.0944$. Certainty of incorrect predictions is more scattered and lower on average, with $\mu = 0.5631$ and $\sigma = 0.1596$. Therefore it is reasonable to conclude that correct classifications are significantly more certain than incorrect ones, i.e., hypothesis 2 is true.

9.2 Intra-model classification

The aim of intra-model recognition is to classify individual cameras inside a camera model. Since the classifying method depends on the sensor imperfections, this experimentation will tell whether the manufacturing processes cause distinguishable variations also in each individual camera.

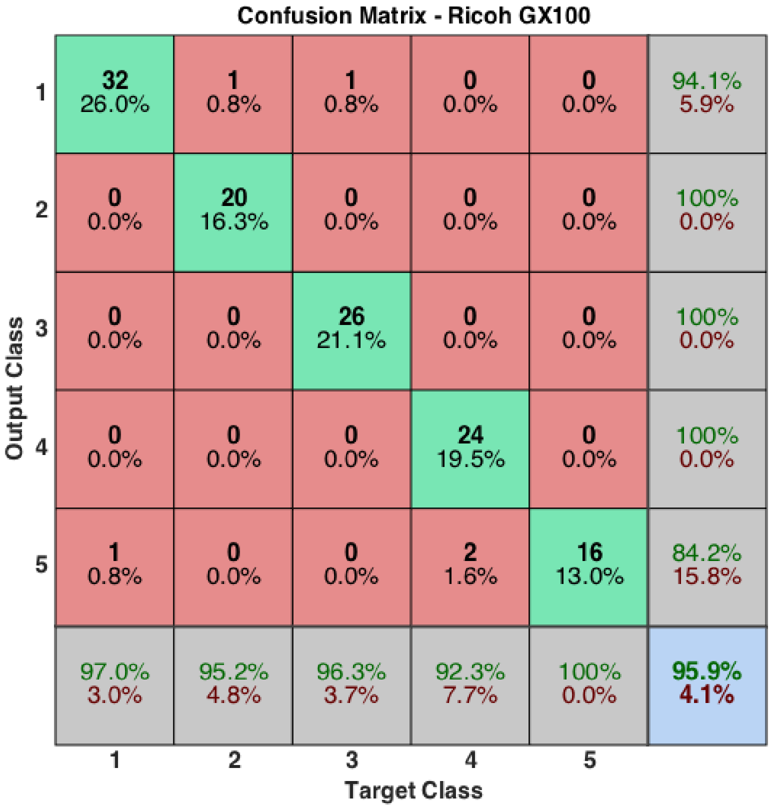
Three camera models were chosen for the experimentation: Ricoh GX100 (5 cameras), Sony DSC-T77 (5 cameras) and Samsung NV15 (3 cameras). Intra-model classification was tested only with wavelet filtering $N_{ref} = 60$ method, since it yielded best results in the previous experimentation. Table 9.3 shows the number off individual cameras for each model and number of images for each camera.

Manufacturer	Model	Individuals	Camera	Images
Ricoh	GX100	5	1	93
			2	81
			3	87
			4	86
			5	76
Samsung	NV15	3	1	130
			2	127
			3	123
Sony	DSC-T77	4	1	90
			2	84
			3	89
			4	88
Total		**12**		**1154**

Table 9.3: Number of cameras in intra-model recognition experimentation

The results for intra-model identification are collected in three confusion matrices:

- Figure 9.10 shows the results for the Ricoh GX100 cameras
- Figure 9.11 shows the results for the Samsung NV15 cameras
- Figure 9.12 shows the results for the Sony DSC-T77 cameras



Confusion Matrix - Ricoh GX100

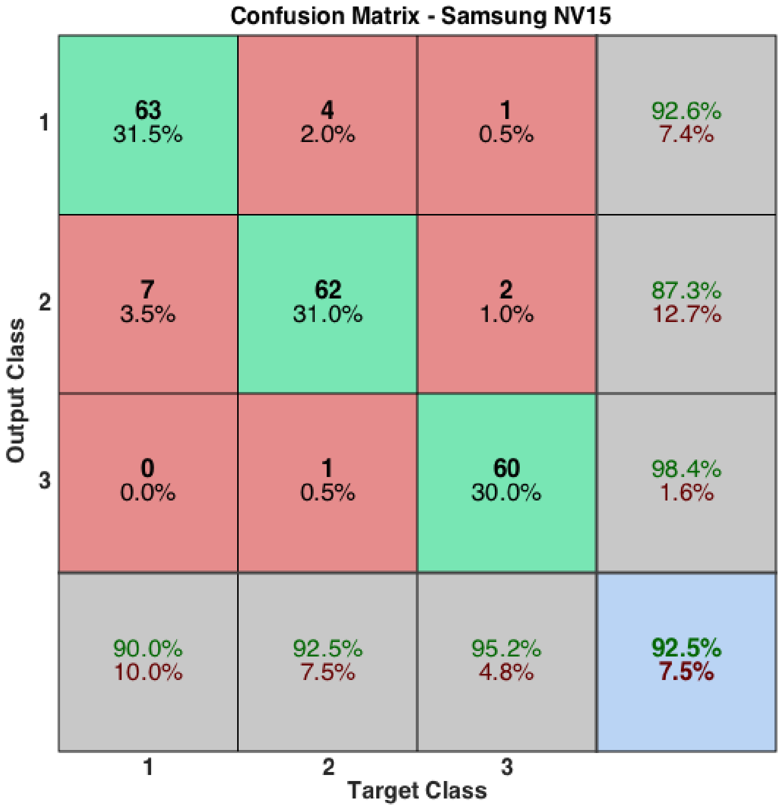
Output Class \ Target Class	1	2	3	4	5	
1	32 26.0%	1 0.8%	1 0.8%	0 0.0%	0 0.0%	94.1% 5.9%
2	0 0.0%	20 16.3%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
3	0 0.0%	0 0.0%	26 21.1%	0 0.0%	0 0.0%	100% 0.0%
4	0 0.0%	0 0.0%	0 0.0%	24 19.5%	0 0.0%	100% 0.0%
5	1 0.8%	0 0.0%	0 0.0%	2 1.6%	16 13.0%	84.2% 15.8%
	97.0% 3.0%	95.2% 4.8%	96.3% 3.7%	92.3% 7.7%	100% 0.0%	**95.9%** **4.1%**

Figure 9.10: Intra-model classification: confusion matrix - Ricoh

The recognition results for Ricoh cameras are fairly good: 95.9%, as shown in figure 9.10. The misclassifications are so rare, that it is impossible to draw further conclusions whether there are camera pairs that are mutually confused with each other. All the cameras have one or two misclassifications, and only camera 5 is always recognized correctly. However, the percentages of correct recognitions are fairly high, and it is plausible that they are due to the randomized split between reference and training images, and have no statistical significance.

Identification performance for Samsung NV15 cameras is a little bit worse than for Ricoh GX100 cameras, i.e., 92.5%, as can be seen in figure 9.11. The worst performance is with camera 1, only 90.0%, but again, it is not likely that this result has any statistical significance either, since the misclassifications are quite evenly spread throughout the confusion matrix.

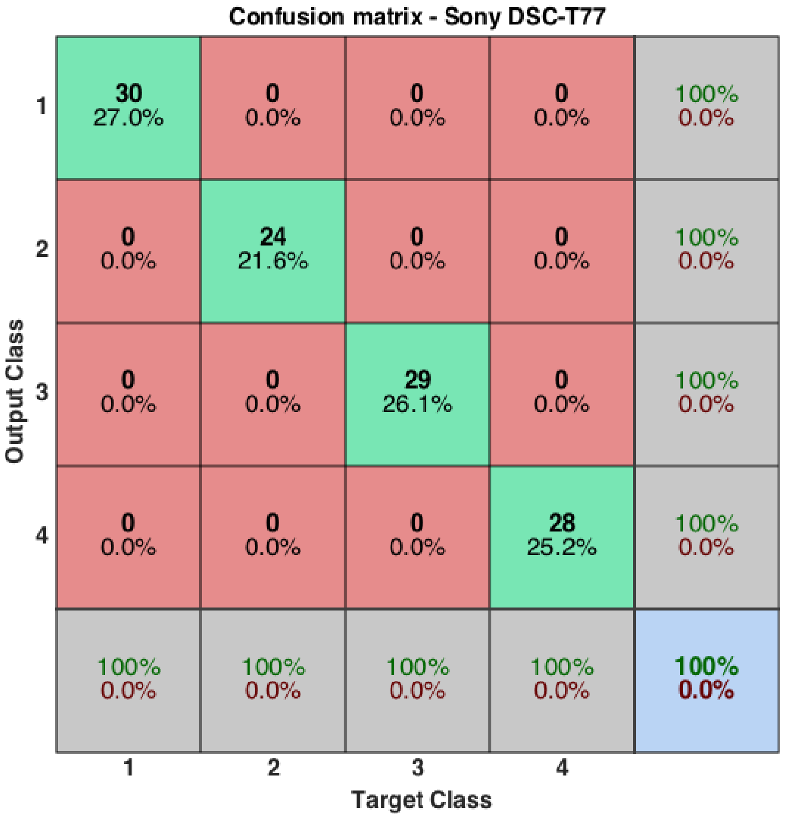
The identification of Sony DSC-T77 cameras yields a perfect result, i.e., all individual cameras are recognized, as shown figure 9.12.



Confusion Matrix - Samsung NV15

Output Class \ Target Class	1	2	3	
1	63 31.5%	4 2.0%	1 0.5%	92.6% 7.4%
2	7 3.5%	62 31.0%	2 1.0%	87.3% 12.7%
3	0 0.0%	1 0.5%	60 30.0%	98.4% 1.6%
	90.0% 10.0%	92.5% 7.5%	95.2% 4.8%	92.5% 7.5%

Figure 9.11: Intra-model classification: confusion matrix - Samsung



Confusion matrix - Sony DSC-T77

Output Class \ Target Class	1	2	3	4	
1	30 27.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
2	0 0.0%	24 21.6%	0 0.0%	0 0.0%	100% 0.0%
3	0 0.0%	0 0.0%	29 26.1%	0 0.0%	100% 0.0%
4	0 0.0%	0 0.0%	0 0.0%	28 25.2%	100% 0.0%
	100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%

Figure 9.12: Intra-model classification: confusion matrix - Sony

These results show that it is quite possible to discern between individual cameras, provided that they are from the same manufacturer and of the same model. However, that must be a priori knowledge, since it is not easy to create model-specific prototype fingerprints, as can be seen in one-tiered approach and model specific fingerprint experimentations in the next two chapters. As a summary, hypothesis 1 is true, i.e., the method gives ≥ 90% accuracy for the classification.

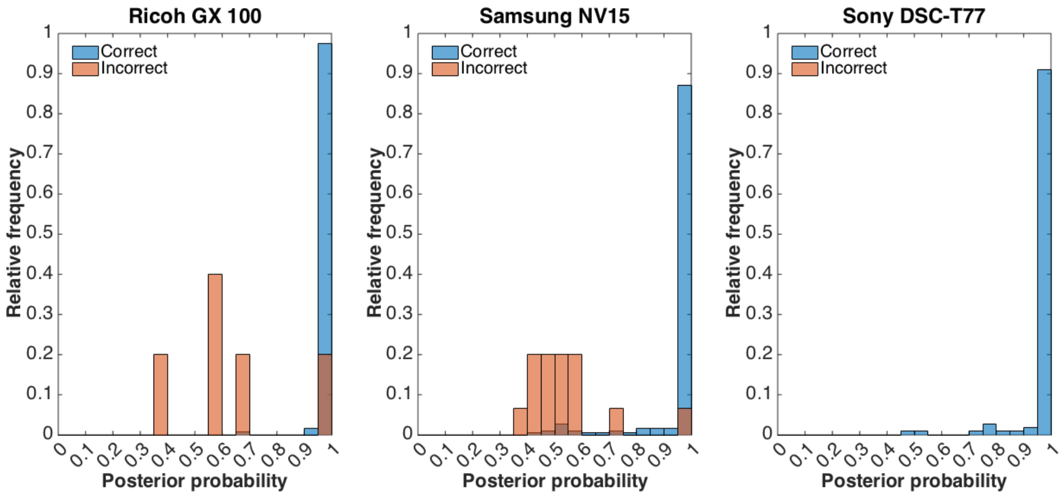


Figure 9.13: Inter-model classification: posterior probabilities for each model and camera

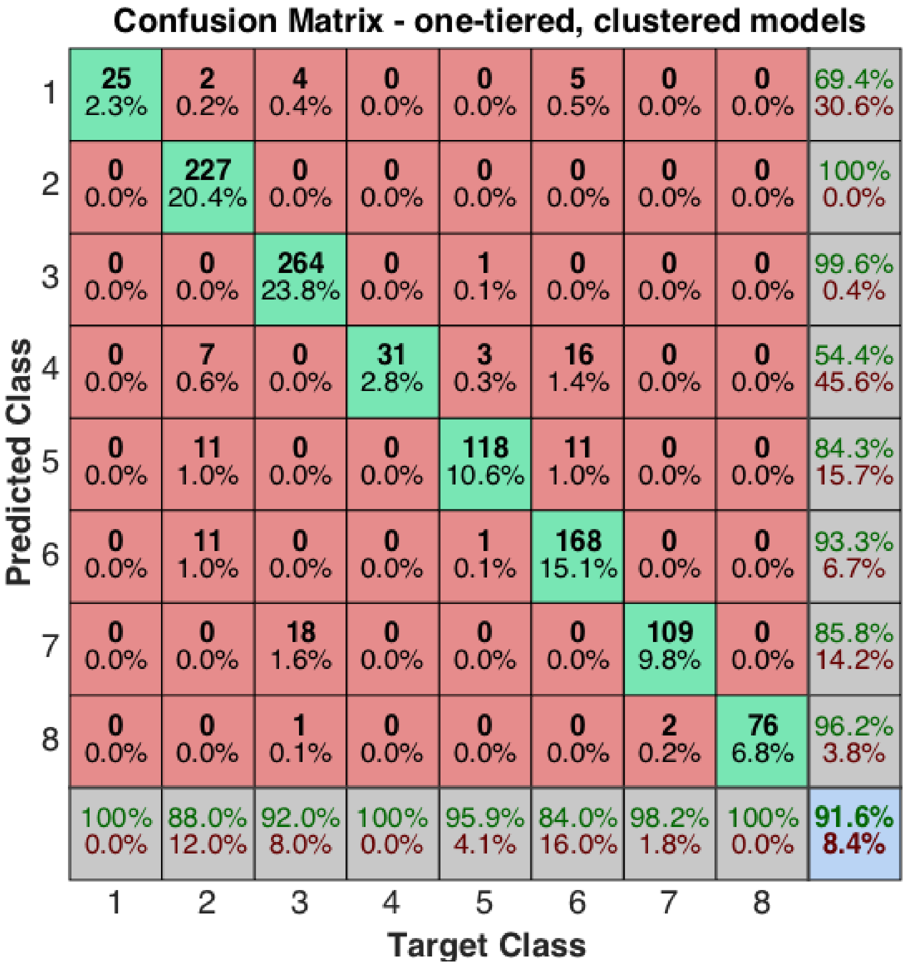
Figure 9.13 shows the relative frequencies of the posterior probabilities for correct and incorrect classifications. As can be seen, correct classifications are usually certain, i.e., the posterior probability is close to 1.0, and the incorrect classifications have lower posterior probabilities. In the case of Sony DSC-T77 camera, there were no incorrect classifications, as can be seen in the confusion matrix in figure 9.12. As can be seen, the distributions of posterior probabilities of correct and incorrect classifications are clearly different, which means that hypothesis 2 is also true in the case of intra-model classification.

9.3 One-tiered approach

This experimentation shows what happens when the distinction between inter-model and intra-model identification is not made. The problem is more complex than the ones in the previous experimentations, as the identification system has to be able to utilize both inter- and intra-model specific features.

Overall, the identification accuracy is 921 images out of 1111, i.e., 82.9%. However, as can be seen in figure 9.15, some of the identification results for clusters around the model, for example for Panasonic DMC-FZ50. The clusters indicate that the inter-model identification works, i.e., the camera model is recognized correctly, but the inter-model variations are too small to make a distinction between individual cameras.

Figure 9.14 shows the results for model recognition in the one-tiered scheme. The results are calculated by combining the results for the individual cameras to



Confusion Matrix - one-tiered, clustered models

Predicted Class \ Target Class	1	2	3	4	5	6	7	8	
1	25 2.3%	2 0.2%	4 0.4%	0 0.0%	0 0.0%	5 0.5%	0 0.0%	0 0.0%	69.4% 30.6%
2	0 0.0%	227 20.4%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
3	0 0.0%	0 0.0%	264 23.8%	0 0.0%	1 0.1%	0 0.0%	0 0.0%	0 0.0%	99.6% 0.4%
4	0 0.0%	7 0.6%	0 0.0%	31 2.8%	3 0.3%	16 1.4%	0 0.0%	0 0.0%	54.4% 45.6%
5	0 0.0%	11 1.0%	0 0.0%	0 0.0%	118 10.6%	11 1.0%	0 0.0%	0 0.0%	84.3% 15.7%
6	0 0.0%	11 1.0%	0 0.0%	0 0.0%	1 0.1%	168 15.1%	0 0.0%	0 0.0%	93.3% 6.7%
7	0 0.0%	0 0.0%	18 1.6%	0 0.0%	0 0.0%	0 0.0%	109 9.8%	0 0.0%	85.8% 14.2%
8	0 0.0%	0 0.0%	1 0.1%	0 0.0%	0 0.0%	0 0.0%	2 0.2%	76 6.8%	96.2% 3.8%
	100% 0.0%	88.0% 12.0%	92.0% 8.0%	100% 0.0%	95.9% 4.1%	84.0% 16.0%	98.2% 1.8%	100% 0.0%	91.6% 8.4%

Figure 9.14: One-tiered approach: confusion matrix for the clustered cameras

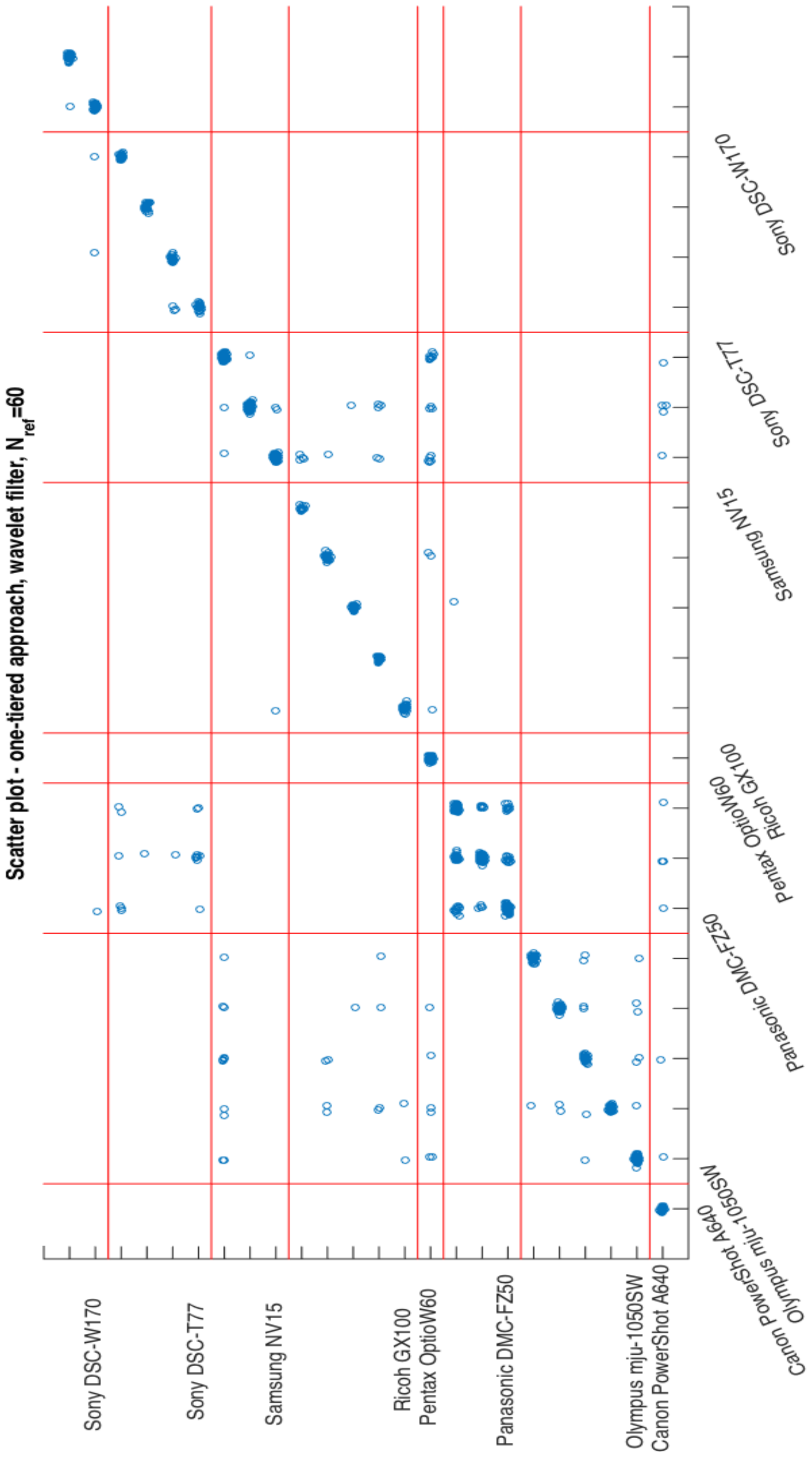


Figure 9.15: One-tiered approach, target vs. predicted classes

one model cluster. The accuracy is 91.6%, which is almost the same as for the inter-model identification experimentation, i.e., 93.2%.

As can be seen, the criterion for hypothesis 1 is not fulfilled neither in the case of simple one-tiered identification scheme, nor in the case of combining individual cameras of the same model into clusters. Since the naïve approach gives lower identification results than the clustered one, the results imply that the inter-model variations override the intra-model ones, and therefore the one-tiered approach is not feasible.

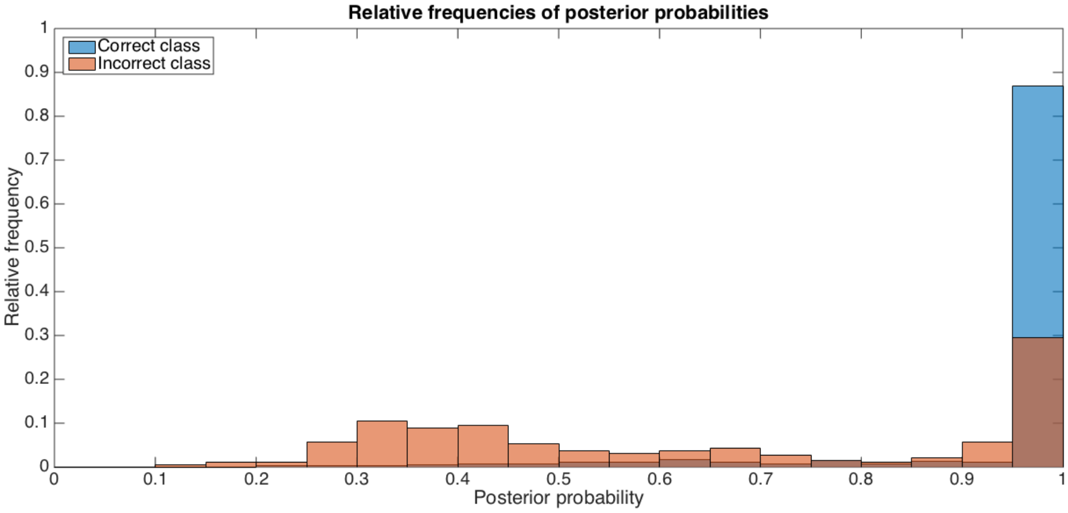


Figure 9.16: One-tiered approach: relative frequencies of posterior probabilities

Figure 9.16 shows the relative frequencies of posterior probabilities in case of both correct and incorrect classifications. As can be seen, the distributions are clearly different, i.e., all the decisions that lead to correct classifications are very certain with $\mu = 0.9533$ and $\sigma = 0.1344$, while the incorrect classifications are fairly uncertain with $\mu = 0.6470$ and $\sigma = 0.2838$. It is to be noted, however, that posterior probabilities of correct classifications have much larger standard deviation than in the case of pure inter-model classification experimentation, the latter being $\sigma = 0.0944$. Therefore hypothesis 2 is true, i.e., the distributions are significantly dissimilar.

9.4 Existence of model specific fingerprints

An interesting question is whether there are camera model specific fingerprints, i.e., whether it is possible to use data from known cameras to create a fingerprint that allows for recognizing the model of

The hypothesis is that when images from several cameras of same model are combined, the individual differences between cameras are small enough to allow for model specific information emerge in the reference patterns. The process that is used to generate the reference patterns is similar to the recognition of individual cameras, except that there is a fixed amount of images from each of the training cameras:

1. Select all but one cameras from a model

2. Randomly select N_{ref} training images from each camera
3. Generate reference patterns by combining the training images of each model
4. Calculate correlations for training images
5. Train bayesian classifier with correlations
6. Calculate correlations with fingerprints for the images from the remaining camera
7. Use trained classifier to classify the images from the remaining camera

Table 9.4 shows the cameras used in this experimentation. The green colored cells mark the camera that is used for testing the system. The column "Cameras" refers to the number of cameras for each model, of which one is used for testing and the rest for training. The column "Training" refers to the size of training pool of which N_{ref} images are chosen for each camera, i.e., all the training images are not used. The column "Testing" refers to the amount of testing images, which will all be used of course in the testing phase.

Camera	**Model**	**Cameras**	**Training**	**Testing**
1	Olympus mju-1050 SW	5	454	104
2	Panasonic DMC-FZ50	3	338	129
3	Ricoh GX 100	5	347	76
4	Samsung NV15	3	257	123
5	Sony DSC-T77	4	263	88
Total		**20**	**1659**	**520**

Table 9.4: Cameras in model specific fingerprint experimentation

As can be seen in figure 9.17, the results are only mediocre at best. The overall accuracies 47.1% and 49.0% for $N = 30$ and $N = 60$ training images, respectively. As can be seen, the system can identify only the model 1, i.e., *Olympus mju–1050 SW*, with a decent accuracy of 80.8%. The performance for camera 4, Samsung NV15, is the worst, with recognition accuracy of only 8.9%. Of the predictions, the images are most often correctly attributed to class 2 and 4, with accuracy of 72.4% and 78.6%. Of predictions, the worst accuracy is with camera 3 - the accuracy of the predictions is only 29.5%.

It is not known at this stage what the cause of the problem is. Inter- and intra-model recognition perform well for both the cameras 1 and 4, and does not give any indication that the accuracy in the case of model specific fingerprints should be this low. Moreover, as can be seen in previous chapters, wavelet filtering with $N_{ref} = 60$ gives excellent recognition accuracy with all the cameras. It is possible to still increase the number of reference images per camera, but as can be seen, the increase from 30 to 60 images does not improve the results sufficiently. Thus, as can be seen, the criteria of hypothesis 1 are not fulfilled, and thus hypothesis 1 is not valid in the case of model specific fingerprints.

Figures 9.18 and 9.19 show the relative frequencies of the posterior probabilities. As can be seen, the correct predictions are very certain only 21.6% and 31.4% of time

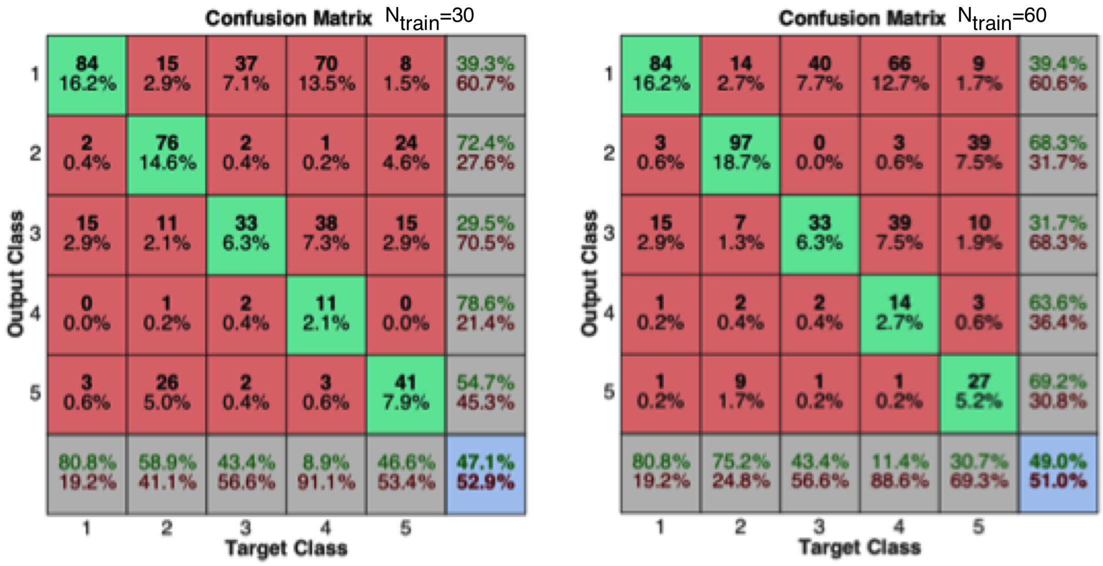


Figure 9.17: Prototype cameras: confusion matrix

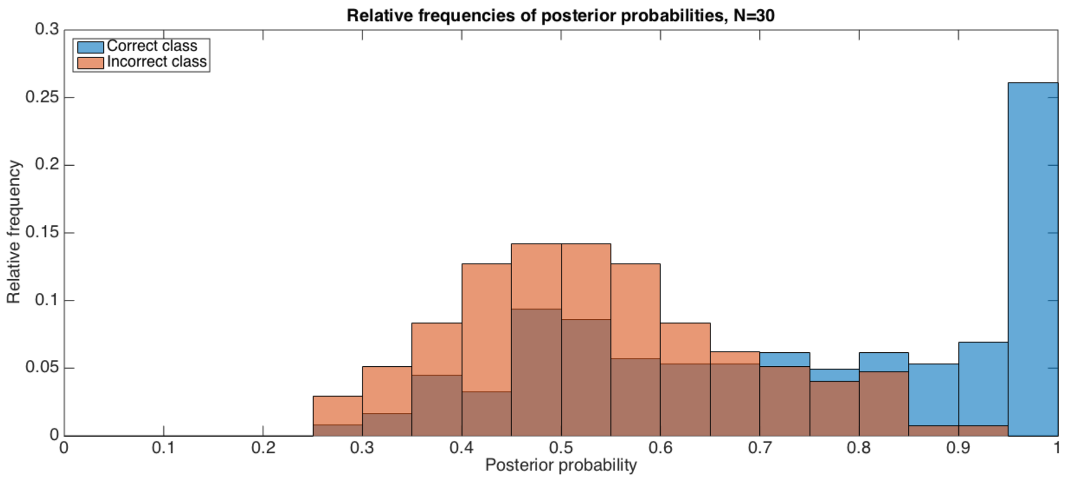


Figure 9.18: Prototype cameras: relative frequencies of posterior probabilities, $N = 30$

for $N = 30$ and $N = 60$. Wrong predictions are not very certain. However, there is a high spread in the certainty of correct predictions when compared with previous experimentations, with $\mu = 0.7312$ and $\sigma = 0.2141$ in the case of 60 training images from each camera. When compared with $\mu = 0.9698$ and $\sigma = 0.0944$ from inter-model classification experiment, the difference is extremely high, which means that the predictions in the case of this experimentation are much more uncertain. Even though one-tiered approach experimentation gave high percentage on model hits, the discrepancy can be explained easily: in the one-tiered approach experimentation all the cameras were used for training, unlike in this experimentation, where one camera from each model was a blind source.

There is a strong similarity between the the distributions of the posterior probabilities of correct and incorrect classifications, If the tail of the relative frequencies of correct classifications is disregarded, the distributions overlap. In the inter-model

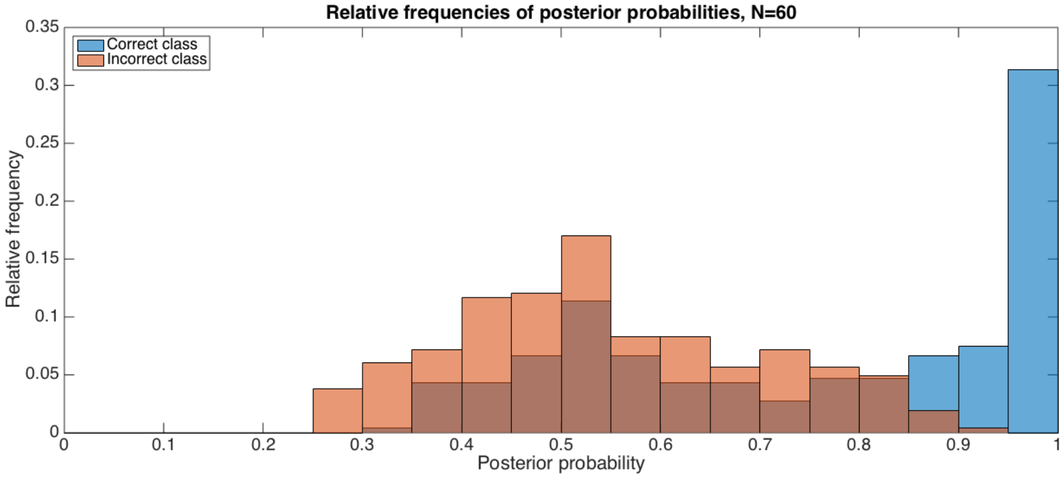


Figure 9.19: Prototype cameras: relative frequencies of posterior probabilities, $N = 60$

experimentation, the correct predictions were very certain and incorrect predictions were uncertain. In this case the distinction cannot be made. Therefore hypothesis 2 is invalid.

Further analysis of the existence of model specific fingerprints is not in the scope of this thesis. However, it certainly is an interesting topic for further work.

Chapter 10

Conclusions and further work

Two hypotheses were to be validated or rejected in this work:

1. Using multivariate multiclass bayesian classifier on the correlation vectors between image noise residual and image fingerprint provides reasonably accurate image source identification method.
2. The posterior probabilities for correct and incorrect classifications are from different kinds of distributions.

In the experimentation plan, there are four experimentations, namely *inter-model classification* (chapter 9.1), *intra-model classification* (chapter 9.2, *one-tiered approach* (chapter 9.3) and *model specific fingerprints* (chapter 9.4). The validity of hypotheses on these experimentations are summarized in table 10.1. As a recapitulation, *inter-model classification* refers to a method that divides images of a camera from each model to training and test set, to learn and exploit the differences between camera models. *Intra-model classification* refers to discerning between cameras of same model. *One-tiered approach* refers to first dividing the images of all the cameras into training and test set, and then inserting all the test images into the same pool. *Model-specific fingerprints* utilizes one camera from each model to generate a fingerprint that is used to recognize the images from the other cameras of the same model.

As can be seen, hypothesis 1 was valid for the inter-model and intra-model recognition experimentations, even though the recognition accuracy was a bit below the set threshold (95%) on one camera model. Hypothesis 1 was rejected in the latter two experimentations: one-tiered approach gave classification accuracy of 91.6%, and model specific fingerprint experimentation gave the accuracy of ca. 49%. It seems reasonable that the results of the first experimentation could be improved sufficiently, but the model specific fingerprint experimentation is probably a hopeless case with the methods used in this thesis.

Hypothesis 2, which measured the stability of the recognition process, was valid for all but the last experimentation. As can be seen in the table, the mean and standard deviation of the distributions of posterior probabilities are clearly different. Also, the posterior probabilities of the correct classifications are high > 0.95 for almost all samples, and thus the classifications are certain, i.e., the process is stable. The posterior probabilities for incorrect classifications are fairly low, and resemble

Experimentation	**H**ypothesis 1		**H**ypothesis 2	
Inter-model	**V**alid	Accuracy with wavelet filtering and 60 images 96.1% >95%	**V**alid	The shape, mean and variance of the posterior probability distributions differ significantly
Intra-model	**V**alid	Recognition accuracy between 92.5% and 100%	**V**alid	The shape, mean and variance of the posterior probability distributions differ significantly
One-tiered	**R**ejected	Recognition accuracy 91.5%	**P**lausible	The shape, mean and variance of the posterior probability distributions differ, but some wrong decisions have high posterior probability
Model specific fingerprints	**R**ejected	Accuracy only ca. 49%	**R**ejected	The distributions have similar characteristics of gaussianity, and statistically similar parameters

Table 10.1: Validity of hypotheses in the experimentations

more a gaussian distribution, compared to the probabilities of correct classifications, which appear as a spike near 1.0. The standard deviations of the posterior probabilities in correct classification are fairly low, i.e., the posterior probabilities have almost no outliers. As expected, the classification process for model specific fingerprints is not stable - the results are almost random, which causes rejection of hypothesis 2 in that case.

Viability of the methods can be assessed with table 10.1. As can be seen, *inter-model classification* has the best performance, with 96.1% accuracy using 60 training images, and the posterior probability distribution, i.e., the certainty of classifications is discernibly different between correct and wrong classifications. Almost all certainties are over $P(\omega_{corr}|\chi) > 0.95$, i.e., if image is classified correctly, the certainty of the decision very high. Instead, if the image is classified incorrectly, the decision is not a certain one - the certainty of $P(\omega_{wrong}|\chi)$ is centered around 0.5 and has a gaussian form with no outliers near 1. Thus, the inter-model classification satisfies both hypothesis 1 and hypothesis 2.

Also *intra-model classification* performs quite well. There were three camera models, i.e., Ricoh GX 100, Samsung NV15 and Sony DSC-T77. The accuracy varies between 92.5% (Samsung NV15) and 100% (Sony DSC-T77). Despite this variation the intra-model classification is deemed to have sufficient accuracy, and therefore satisfies the hypothesis 1. The certainties are bit more complex; the camera manufactured by Ricoh presented some certainties above 95% also for wrong predictions, even though most certainties in the case of wrong predictions fell on gaussian curve with mean around 0.5. However, since the number of misclassified images was rather low, it is not possible to analyze the situation further. The certainties for correctly classified images from the Ricoh camera were over 0.95. The certainties of misclassifications for the Samsung camera fell on the gaussian curve with mean of 0.5, and correct classifications on a strong spike with $P > 0.95$. There were no misclassifications for the Sony camera, and large majority of the classifications had $P > 0.95$. Therefore, hypothesis 2 in the case of intra-model classification is considered to be satisfied.

The *one-tiered experimentation* aspired to combine the two preceding approaches. As can be seen in table 10.1, hypothesis 1 had to be rejected outright, since the accuracy was only 91.5%, which is unacceptable for a system intended to be used in forensics. Strictly speaking, also the hypothesis 2 should be rejected, since the distributions of posterior probabilities partly overlap, i.e., also the wrong decisions have high certainties.

Model-specific fingerprints experimentation had the worst performance and stability. As an approach it could be called a catastrophe, since the accuracy was only 49%, i.e., hypothesis 1 was rejected, and the distributions of certainty overlapped strongly, i.e., hypothesis 2 was rejected. However, as usually happens in science, a lot can be learned from the experimentation. First and foremost, one-tiered and model specific fingerprint experimentations imply strongly that choosing only one camera of a particular model in the inter-model experimentations was a right choice. As can be seen, one camera cannot represent the whole model, i.e., act as a prototype, and putting all the images in the same pool is not a good choice - the noise resolution is not sufficient. This experimentation did not invalidate the idea of prototype cameras; instead, it set some boundary conditions for the further methodological development.

Therefore, as a main conclusion of this work, these experimentations have shown that using two-tiered inter- and intra-model classification system with wavelet filtering and ca. 60 training images can be used in image source recognition even in forensics. The hypotheses were rather ambitious, requiring high accuracy and stability, and were fulfilled in the crucial experimentations. It can be easily said that the methods used in the first and second experimentations exceeded expectations.

These experimentations have also shown that it is possible to perform source camera identification with mathematically and computationally simple methods. Wavelet filtering notwithstanding, the mathematical methods are based on simple matrix algebra, which can be easily optimized with well known numerical algorithms. From an engineering point of view, the methods are very feasible – images have to be filtered only once and can be saved after, thus eliminating one time consuming step from subsequent runs. Also, distributed computer architecture can be exploited in order to further speed up the process.

As a final note, it must be remembered that these methods can be used only if there is a priori knowledge on the cameras. These methods do not allow for blind source camera recognition; however, with the help of existing forensic image databases, it is possible to extends the feasibility of this work considerably by acquiring the necessary a priori knowledge by other forensic means.

10.1 Further work

One-tiered approach and model specific fingerprint experimentations set a clear path for further methodological development. The experimentations failed to satisfy the hypotheses, but as always, post mortem examination produces interesting results.

The impact of the amount of training images was clearly visible, but not as important as could be thought. For example, in the inter-model experimentation it was shown that the difference between training image set sizes $N_{train} = 50$ and $N_{train} = 60$ with wavelet filtering was only 2.9 percentage points. The experimentation allowed to select the filtering method and training image set sizes for further experimentations. However, the impact of changing the training image set sizes is not known, but taken from literature (Lukáš et al., 2006). Changing the set size is not as straightforward as it sounds, since as always with pattern recognition, there is a risk of under- or overlearning the patterns. The impact of the set size should be examined further.

Using other filtering methods to elicit noise patterns has to be examined, since the failure of the one-tiered and model-specific fingerprint approaches to satisfy the hypotheses can be related to an unsuitable filtering method. Using a wavelet filter is not a wrong choice as such, but other filtering methods or cascades of filters could perform better. Of the other filtering methods, e.g. partial differential equations show some promise (You and Kaveh, 2000) (Liu et al., 2011).

When the impact of training image set sizes and filter construction are known, the possibility to create model-specific fingerprints should be examined. These fingerprints would allow to detect the camera model and manufacturer even when the source camera is not available. Instead of finding the exact camera, this would allow for quicker sifting through suspect cameras.

One-tiered approach is important to allow for speeding up the recognition process, and to do minor enhancements on the accuracy and stability of inter- and intra-model approaches.

Other classification methods besides the Bayesian should be further experimented with, e.g., naïve comparison of correlations, neural networks and regression analysis. Some of these were presented in this thesis, but they are not sufficiently studied. It should be also studied how to diminish the probability of false positives, since those can have dire consequences for the camera owner.

In this work, it was assumed that all the images come directly from the camera unedited, and only models with the same image sizes were considered. This presents three interesting questions:

- Is there a image size scaling algorithm that can preserve enough information of the noise pattern so that also cameras with different image sizes?
- Is there a suitable method to find the correct position for a cropped image in the noise pattern of a camera?
- Is there a way to restore the original image sufficiently if image enhancement algorithms have been used?

Answering these questions would clearly advance the possibilities of the source camera identification system presented in this thesis. Image restoration and enhancement algorithm detection are largely on-going work, which of course implies carrying out comprehensive literature studies.

Image compression sets quite a few challenges for source identification. Since lossy image compression specifically reduces small variations, e.g. noise. The higher the compression ratio is, the less image noise is preserved. Also, low resolution is a challenge to be addressed. (Alles et al., 2009)

There are of course two parts when information, especially noise patterns, is considered. First is self-evidently storing, and the second is finding it, i.e., indexing. Since the noise patterns have the exact size and color depth of the original images, finding a sufficient storage is not a problem nowadays. However, indexing is. Some approaches for indexing images based on features have been presented, and the first such to continue the work in this thesis would be using pyramid matching kernels, as shown in (Grauman) (Grauman and Darrell, 2007) (Grauman and Darrell, 2005). Of course, indexing presents another interesting option – could two noise patterns be compared with their indexes instead of direct methods. Image indexing is a heavily researched area nowadays, and is likely to contribute to source camera identification.

This thesis is an example of applying supervised learning, i.e., the source of images, and thus the origin of the noise pattern, is known in the learning phase. Unsupervised learning, e.g. clustering methods, are a topic of interest, since they allow for grouping data with no a prior knowledge of the source.

References

Alles, E. J., Geradts, Z. J. M. H., and Veenman, C. J. (2009). Source Camera Identification for Heavily JPEG Compressed Low Resolution Still Images. *Journal of Forensic Sciences*, 54(3):628–638.

Alpaydin, E. (2010). *Introduction to Machine Learning.* MIT Press, Cambridge, MA, second edition.

Baar, T., van Houten, W., and Geradts, Z. (2012). Camera identification by grouping images from database, based on shared noise patterns. *arXiv.org*.

Chen, M., Fridrich, J., Goljan, M., and Lukáš, J. (2008). Determining Image Origin and Integrity Using Sensor Noise. *Information Forensics and Security, IEEE Transactions on*, 3(1):74–90.

Donoho, D. L. (1995). De-noising by soft-thresholding. *Information Theory, IEEE Transactions on*, 41(3):613–627.

European Central Bank (2013). Europa series, 5 euro banknote.

Frery, A. C. and Perciano, T. (2013). Filters in the Image Domain. In *Introduction to Image Processing Using R*, pages 59–75. Springer London, London.

Gloe, T. and Böhme, R. (2010). The 'Dresden Image Database' for benchmarking digital image forensics. In *the 2010 ACM Symposium*, page 1584, New York, New York, USA. ACM Press.

Goljan, M. and Fridrich, J. (2008). Camera identification from cropped and scaled images. In *SPIE , Security, Forensics, Steganography, and Watermarking of Multimedia Contents X*, pages 68190E–68190E–13. International Society for Optics and Photonics.

Grauman, K. *Matching Sets of Features for Efficient Retrieval and Recognition.* PhD thesis, Massachusetts Institute of Technology.

Grauman, K. and Darrell, T. (2005). The pyramid match kernel: discriminative classification with sets of image features. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, pages 1458–1465.

Grauman, K. and Darrell, T. (2007). The Pyramid Match Kernel: Efficient Learning with Sets of Features. *The Journal of Machine Learning Research*, 8.

Gunturk, B. K., Glotzbach, J., Altunbasak, Y., Schafer, R. W., and Mersereau, R. M. (2005). Demosaicking: color filter array interpolation. *Signal Processing Magazine, IEEE*, 22(1):44–54.

Huang, G.-B., Zhu, Q.-Y., and Siew, C.-K. (2006). Extreme learning machine: Theory and applications. *Neurocomputing*, 70(1-3):489–501.

Hullin, M., Eisemann, E., Seidel, H.-P., and Lee, S. (2011). Physically-based real-time lens flare rendering. *SIGGRAPH '11: SIGGRAPH 2011 papers*, 30(4):1.

Komprobst, P., Deriche, R., and Aubert, G. (1997). Image coupling, restoration and enhancement via PDE's. In *International Conference on Image Processing*, pages 458–461. IEEE.

Liu, X., Huang, L., and Guo, Z. (2011). Adaptive fourth-order partial differential equation filter for image denoising. *Applied Mathematics Letters*, 24(8):1282–1288.

Lukáš, J., Fridrich, J., and Goljan, M. (2006). Digital camera identification from sensor pattern noise. *IEEE Transactions on Information Forensics and Security*, 1(2):205–214.

Mallat, S. G. and Mallat, C. (1999). *Wavelet Tour of Signal Processing.* Academic Press, London, UK, 2nd edition.

Memane, T. and Ruikar, S. D. (2014). Selection of wavelet for satellite image compression using picture quality measures. In *Communications and Signal Processing (ICCSP), 2014 International Conference on*, pages 1003–1006. IEEE.

Mitra, S. K. (2011). *Digital Signal Processing.* A Computer-based Approach. McGraw-Hill.

Mojsilović, A., Popović, M. V., and Rackov, D. M. (2000). On the selection of an optimal wavelet basis for texture characterization. *IEEE Transactions on Image Processing*, 9(12):2043–2050.

Ramanath, R., Snyder, W. E., Yoo, Y., and Drew, M. S. (2005). Color image processing pipeline. *IEEE Signal Processing Magazine*, 22(1):34–43.

Rocha, A., Scheirer, W., Boult, T., and Goldenstein, S. (2011). Vision of the unseen. *ACM Computing Surveys*, 43(4):1–42.

Tan, H. G. R., Tan, A. C., Khong, P. Y., and Mok, V. H. (2007). Best Wavelet Function Identification System for ECG signal denoise applications. In *Intelligent and Advanced Systems, 2007. ICIAS 2007. International Conference on*, pages 631–634.

Tang, Y. Y. (2009). *Wavelet Theory Approach to Pattern Recognition.* World Scientific Publishing Co., SGP.

Thuillard, M. (2001). *Wavelets in Soft Computing.* World Scientific, River Edge, NJ, USA.

Tsai, D.-Y., Lee, Y., Sekiya, M., Sakaguchi, S., and Yamada, I. (2002). A method of medical image enhancement using wavelet analysis. In *Signal Processing, 2002 6th International Conference on*, pages 723–726 vol.1. IEEE.

Wang, L., Huang, Y., Luo, X., Wang, Z., and Luo, S. (2011). Image deblurring with filters learned by extreme learning machine. *Neurocomputing*, 74(16):2464–2474.

You, Y. L. and Kaveh, M. (2000). Fourth-order partial differential equations for noise removal. *Image Processing, IEEE Transactions on*, 9(10):1723–1730.

Glossary

Bayesian classification calculates the posterior probability for a sample belonging to class k, i.e., $P(\omega_k|\chi)$, where χ is the sample, using Bayes's rule:

$$P(\omega_k|\chi) = \frac{P(\chi|\omega_k)P(\omega_k)}{P(\chi)}$$

The sample is then assigned to class k with highest posterior probability $P(\omega_k|\chi)$. See *linear regression*

biorthogonal wavelet base for practical purposes, biorthogonal wavelet base has separate scaling and mother wavelet functions for decomposition and reconstruction, designated as ϕ_d and ψ_d (decomposition), and ϕ_r and ψ_r (reconstruction), see *orthogonal wavelet base*

blind source may refer to

1. an unknown signal source, e.g., a camera that the forensic scientist has no access to, and has no knownledge of the properties of the device, or even how many devices there are
2. situation in which there are a large set of images, but it is not known how many cameras have been used and which images are from which cameras

convolution discrete time convolution refers to sliding one sequence $h[n]$ over another sequence $x[n]$ to produce result vector $y[n]$. It is defined as sum

$$y[n] = \sum_{k=-\infty}^{\infty} x[k]h[n-k]$$

dark current refers to the rate of electrons accumulating in each sensor pixel due to thermal action caused by the thermal energy inherent to the structure of the sensor and is independent of light falling on it

discrete wavelet transform applying discrete lowpass and highpass analysis filters, defined by wavelet base function, to represent a discrete signal as wavelet coefficients on various levels of detail

downsampling downsampling by factor M means that only every Mth sample of the signal is kept, i.e., $M-1$ samples are dropped between each kept sample; the resultant sampling frequency is $\frac{1}{M}$ of the original signal

Extreme Learning Machine a variation of neural network algorithm that has very fast learning capabilities and is immune to local minima and discontinuities

fast Fourier transform a fast algorithm for estimating the coefficients of Fourier transform for discrete data, e.g., audio or image samples

filter an implementation of a mathematical method that removes noise or other unwanted phenomena from a signal, e.g. an image or a voice recording, most often based on convolution or signal transforms

filter bank is utilized by a wavelet filter to separate different signal frequencies to allow for multi-resolution inspection and filtering of complex data such as images

flat fielding An image taken with a uniformly illuminated imaging sensor, e.g. solid colored and uniformly illuminated surface. All information in the image is a result of PRNU or stochastic noise.

Fourier transform A mathematical transformation technique for mapping a signal in time-amplitude to frequency-power scale to allow for special filtering techniques, see *inverse Fourier transform*

gaussian filtering a fast spatial filter based on computing gaussian coefficient weighted sum of pixel neighbourhood; efficient and efficient for some types of noise, but causes blurring and other artefacts, see *wavelet filter*

inverse discrete wavelet transform applying discrete lowpass and highpass synthesis filters, defined by the wavelet base, on wavelet coefficients to recover the denoised signal

inverse Fourier transform A mathematical transformation technique for mapping a signal represented in frequency-power scale to time-amplitude scale, usually to reverse Fourier Transform, see *Fourier transform*

lens flare scattering and reflections of light from a strong light source in the view, when the light passes the camera optics

linear regression correlations between images and reference patterns are linearly mapped on a line that can be divided into regions denoting the best matching pattern. Linear regression can be either *univariate* or *multivariate*. See *Bayesian classification*

linear time-invariant the response of a system does not change with time shifts or signal amplitude change, i.e., a system $\mathcal{H}$ is linear and time-invariant if two conditions hold:

1. $\mathcal{H}(\alpha\, u[n] + \beta\, v[n]) = \alpha\mathcal{H}(u[n]) + \beta\mathcal{H}v[n])$ (linearity)
2. $\forall\, r : y[n] = \mathcal{H}(x[n]) \Rightarrow\ y[n-r] = \mathcal{H}(x[n-r])$ (time-invariance)

multivariate refers to methods that use an input vector to produce the output value; in the case of camera identification, the output classification is determined by considering the correlations between an image and all of the reference patterns, see *univariate*

neural network machine learning algorithms that can learn complex structures in data, the idea originating from simulating brain neuron cells

orthogonal wavelet base for practical purposes, orthogonal wavelet base uses the same scaling ϕ and mother wavelet ψ functions for both decomposition and reconstruction, see *biorthogonal wavelet base*

reference image an image used for forming the camera fingerprint in the training phase of the source identification system, see *test image*

synchronization fingerprints and the test image have to share same properties with respect to transformations, e.g. rotation, scaling and cropping

test image an image used for testing the source identification system, see *reference image*

univariate refers to methods that use only one input value to produce the output value; in the case of camera identification, the output classification is determined by the correlation of an image with one pattern, see *multivariate*

upsampling upsampling by factor L means that a sequence of zeroes with length $L-1$ is inserted between each consecutive samples of sequence $x[n]$ so that the sample rate of the new sequnce is L times the sampling frequency of the original signal. The upsampled sequence $x_u[n]$ is

$$x_u[n] = \begin{cases} x[\frac{n}{L}] & n = \pm L, \pm 2L \pm 3L, ... \\ 0 & \text{otherwise} \end{cases}$$

wavelet filter a very effective image filter based on discrete wavelet transform (DWT) and filter banks, that can remove noise on various detail levels, see *gaussian filtering*

Index