

Microphone Classification Using Fourier Coefficients

Robert Buchholz, Christian Kraetzer, Jana Dittmann

Otto-von-Guericke University of Magdeburg, Department of Computer Science, PO Box
4120, 39016 Magdeburg, Germany
{robert.buchholz, christian.kraetzer, jana.dittmann}@iti.cs.uni-magdeburg.de

Abstract. Media forensics tries to determine the originating device of a signal. We apply this paradigm to microphone forensics, determining the microphone model used to record a given audio sample. Our approach is to extract a Fourier coefficient histogram of near-silence segments of the recording as the feature vector and to use machine learning techniques for the classification. Our test goals are to determine whether attempting microphone forensics is indeed a sensible approach and which one of the six different classification techniques tested is the most suitable one for that task. The experimental results, achieved using two different FFT window sizes (256 and 2048 frequency coefficients) and nine different thresholds for near-silence detection, show a high accuracy of up to 93.5% correct classifications for the case of 2048 frequency coefficients in a test set of seven microphones classified with linear logistic regression models. This positive tendency motivates further experiments with larger test sets and further studies for microphone identification.

Keywords: media forensics, FFT based microphone classification

1 Motivation

Being able to determine the microphone type used to create a given recording has numerous applications. Long-term archiving systems such as the one introduced in the SHAMAN project on long-term preservation [3] store metadata along with the archived media. Determining the microphone model or identifying the microphone used would be a useful additional media security related metadata attribute to retrieve recordings by.

In criminology and forensics, determining the microphone type and model of a given alleged accidental or surveillance recording of a committed crime can help determining the authenticity of that record. Furthermore, microphone forensics can be used in the analysis of video statements of dubious origin to determine whether the audio recording could actually have been made by the microphone seen in the video or whether the audio has been tempered with or even completely replaced. Also, other media forensic approaches like gunshot characterization/classification [12] require knowledge about the source characteristics, which could be established with the introduced microphone classification approach.

Finally, determining the microphone model of arbitrary recordings can help determine the actual ownership of that recording in the case of multiple claims of

ownership, and can thus be a valuable passive mechanism like perceptual hashing in solving copyright disputes.

The goal of this work is to investigate whether it is possible to identify the microphone model used in making a certain audio recording by using only Fourier coefficients. Its goal is not comprehensively cover the topic, but merely to give an indication on whether such a classification is indeed possible. Thus, the practical results presented may not be generalizable.

Our contribution is to test the feature extraction based only on frequency domain features of near-silence segments of a recording (using two different FFT window sizes (256 and 2048 frequency coefficients) and nine different thresholds for near-silence detection) and to classify the seven microphones using six different classifiers not yet applied to this problem. In this process, two research questions are going to be answered: First, is it possible to classify microphones using Fourier coefficient based features, thereby reducing the complexity of the approach presented in [1]? Second, which classification approach out of a variety (logistic regression, support vector machines, decision trees and nearest neighbor) is the most suitable one for that task? Additionally, first indications for possible dimensionality reduction using principal component analysis (PCA) are given and inter-microphone differences in classification accuracy are mentioned.

The remaining paper is structured as follows: Section 2 presents related work to place this paper in the larger field of study. Section 3 introduces our general testing procedure, while Section 4 details the test setup for the audio recordings and Section 5 details the feature extraction and classification steps. The test results are then shown in Section 6 and are compared with earlier results. The paper is concluded with a summary and an outlook on future work in Section 7.

2 Related Work

The idea of identifying recording devices based on the records produced is not a new one and has been attempted before with various device classes. A recent example for the great variety is the work from Filler et al. [6] as well as Dirik et al. [7] on investigating and evaluating the feasibility of identifying a camera used to take a given picture. Forensics for flatbed scanners is introduced in [8] and Khanna et al. summarize identification techniques for scanner and printing device in [9]. Aside from device identification, other approaches also look into the determination of the used device model, such as done in for cameras models in [10] or performed for handwriting devices in [11]. However, to our knowledge no other research group has yet explored the feasibility of microphone classification.

Our first idea based on syntactical and semantic feature extraction, analysis and classification for audio recordings was introduced in [13] and described a first theoretical concept of a so-called verifier-tuple for audio-forensics. Our first practical results were presented by Kraetzer et al. [1] and were based on a segmental feature extractor normally used in steganalysis (AAST; computing seven statistical measures and 56 cepstral coefficients). The experiments were conducted on a rather small test setup containing only four microphones for which the audio samples were recorded

simultaneously – an unlikely setup for practical applications that limits the generalizability of the results. These experiments also used only two basic classifiers (Naïve Bayes, k-means clustering). The results demonstrated a classification accuracy that was clearly above random guessing, but was still by far too low to be of practical relevance. Another approach using Fourier coefficients as features was examined in our laboratory with an internal study [5] with an extended test set containing seven different microphones. The used minimum distance classifier proved to be inadequate for the classification of high-dimensional feature vectors, but these first results were a motivation to conduct further research on the evaluation of Fourier features with advanced classification techniques. The results are presented and discussed in this paper.

3 Concept

Our approach is to investigate the ability of feature extraction based on a Fourier coefficient histogram to accurately classify microphones with the help of model based classification techniques.

Since Fourier coefficients are usually characteristic for the sounds recorded and not for the device recording it, we detect segments of the audio file that contain mostly noise, and apply the feature extractor only to these segments. The corresponding Fourier coefficients for all those segments are summed up to yield a Fourier coefficient histogram that is then used as the global feature vector.

The actual classification is then conducted using the WEKA machine learning software suite [2]. The microphone classification task is repeated with different classification algorithms and parameterizations for the feature extractor (non-overlapping FFT windows with 256 and 2048 coefficients (therefore requiring 512 and 4096 audio samples per window), and nine different near-silence amplitude thresholds between zero and one for the detection of segments containing noise). With the data on the resulting classification accuracies, the following research questions can be answered:

1. Is it possible at all to determine a microphone model based on Fourier coefficient characteristics of a recording using that microphone?
 2. Which classifier is the most accurate one for our microphone classification setup?
- In addition, we give preliminary results on whether a feature space reduction might be possible and investigate inter-microphone differences in classification accuracy.

4 Physical Test Setup

For the experiments, we focus on microphone model classification as opposed to microphone identification. Thus, we do not use microphones of the same type.

The recordings are all made using the same computer and loudspeaker for playback of predefined reference signals. They are recorded for each microphone separately, so that they may be influenced by different types of environmental noise to differing degrees. While this will likely degrade classification accuracy, it was done on purpose

in order for the classification results to be more generalizable to situations where synchronous recording of samples is not possible.

There are at least two major factors that may influence the recordings besides the actual microphones that are supposed to be classified: The loudspeaker used to play back a sound file in order for the microphone to record the sample again and the microphone used to create the sound file in the first place. We assume that the effect of the loudspeaker is negligible, because the dynamic range of the high quality loudspeaker used by far exceeds that of all tested microphones. The issue is graver for the microphones used to create the sound files, but varies depending on the type of sound file (see Table 1).

Table 1. The eight source sound files used for the experiments (syntactical features: 44.1kHz sampling rate, mono, 16 bit PCM coding, average duration 30s).

File Name	Content
Metallica-Fuel.wav	Music, Metal
U2-BeautifulDay.wav	Music, Pop
Scooter-HowMuchIsTheFish.wav	Music, Techno
mls.wav	MLS Noise
sine440.wav	440Hz sine tone
white.wav	White Noise
silence.wav	Digital silence
vioo10_2_nor.wav	SQAM, instrumental

Some of our sound samples are purely synthetic (e.g. the noises and the sine sound) and thus were not influenced by any microphone. The others are short audio clips of popular music. For these files, our rationale is that the influence of the microphones is negligible for multiple reasons. First, they are usually recorded with expensive, high quality microphones whose dynamic range exceeds that of our tested microphones. And second, the final piece of music is usually the result of mixing sounds from different sources (e.g. voices and instruments) and applying various audio filters. This processing chain should affect the final song sufficiently in order for the effects of individual microphones to be no longer measurable.

We tested seven different microphones (see Table 2). None are of the same model, but some are different models from the same manufacturer. The microphones are based on three of the major microphone transducer technologies.

Table 2. The seven microphones used in our experiments.

Microphone	Transducer technology
Shure SM 58	Dynamic microphone
T.Bone MB 45	Dynamic microphone
AKG CK 93	Condenser microphone
AKG CK 98	Condenser microphone
PUX 70TX-M1	Piezoelectric microphone
Terratec Headset Master	Dynamic microphone
T.Bone SC 600	Condenser microphone

All samples were played back and recorded by each microphone in each of twelve different rooms with different characteristics (stairways, small office rooms, big

office rooms, a lecture hall, etc.) to ensure that the classification is independent from the recording environment.

Thus, the complete set of audio samples consists of 672 individual audio files, recorded with seven microphones in twelve rooms based on eight source audio files. Each file is about 30 seconds long, recorded as an uncompressed PCM stream with 16 bit quantization at 44.1kHz sampling rate and a single audio channel (mono). To allow amplitude-based operations to work equally well on all audio samples, the recordings are normalized using SoX [4] prior to feature extraction.

5 Feature Extraction and Classification

Our basic idea is to classify for each recorded file f the microphones based on the FFT coefficients of the noise portion of the audio recordings. Thus, the following feature extraction steps are performed for each threshold t tested ($t \in \{0.01, 0.025, 0.05, 0.1, 0.2, 0.225, 0.25, 0.5, 1\}$ for $n=256$ and $t \in \{0.01, 0.025, 0.05, 0.1, 0.25, 0.35, 0.4, 0.5, 1\}$ for $n=2048$; cf. Figure 1):

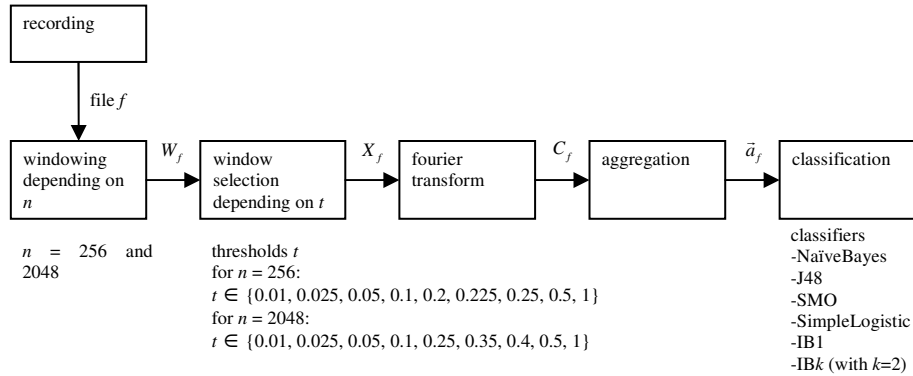


Fig. 1. The classification pipeline.

The feature extractor first divides each audio file f of the set of recorded files F into equally-sized non-overlapping windows W_f with size $2n$ samples ($W_f = \{w_1(f), w_2(f), \dots, w_m(f)\}$; $m = \text{sizeof}(f)/(2n)$). Two different values for n are used in the evaluations performed here: $n=256$ and $n=2048$. For each file f , only those windows $w_i(f)$ ($1 \leq i \leq m$) are selected for further processing where the maximum amplitude in the window does not exceed a variable near-silence threshold t and thus can be assumed to contain no content but background noise. For each n nine different t are evaluated here.

All s selected windows for f form the set X_f ($X_f \subseteq W_f$, $X_f = \{x_1(f), x_2(f), \dots, x_s(f)\}$). These s selected windows are transformed to the frequency domain using a FFT and the amplitude portion of the complex-valued Fourier coefficients is computed. The resulting vector of n Fourier coefficients

(harmonics) for each selected window is identified as $\vec{c}_{fj} = (FC_1, FC_2, \dots, FC_n)_{fj}$ with $1 \leq j \leq s$. Thus, for each file f , a set C_f of s coefficient vectors \vec{c}_{fj} of dimension n is computed. To create a constant length feature vector \vec{a}_f for each f the amplitudes representing the same harmonic in each element in C_f are summed up, yielding an amplitude histogram vector \vec{a}_f of size n . To compensate for different audio sample lengths and differences in volume that are not necessarily characteristics of the microphone the feature vector is normalized as a last step so that its maximum amplitude value is one.

Parameter Considerations. The test setup allows for two parameters to be chosen with some constraints: the FFT window size ($2n$ and has to be a power of two) and the amplitude threshold t (between 0 and 1) that decides whether a given sample window contains mostly background noise and thus is suitable to characterize the microphone. Both need to be considered carefully because their values represent trade-offs:

If t is chosen too low, too few windows will be considered suitable for further analysis. This leads the amplitude histogram to be based on fewer samples and thus will increase the influence of randomness on the histogram. In extreme cases, a low threshold may even lead to all sample windows being rejected and thus to an invalid feature vector. On the other hand, a high t will allow windows containing a large portion of the content (and not only the noise) to be considered. Thus, the influence of the characteristic noise will be reduced, significantly narrowing the attribute differences between different microphones. For the experiments, various thresholds ranging from 0.01 to 1 are tested, with special focus on those thresholds for which the feature extraction failed for few to no recordings, but which are still small enough to contain mostly noise.

For the FFT coefficients, a similar trade-off exists: If the FFT window size is set rather low, then the number n of extracted features may be too low to distinguish different microphones due to the reduced frequency resolution. If the window size is set too high, the chances of the window to contain at least a single amplitude that exceeds the allowed threshold and thus being rejected increases, having the same negative effects as a high threshold. Additionally, since the feature vector size n increases linearly with the window size, the computation time and memory required to perform the classification task increases accordingly. To analyze the effect of the window size on the classification accuracy, we run all tests with $n = 256$ and $n = 2048$ samples. For $n = 2048$, some classifications already take multiple hours, while others terminate the used data mining environment WEKA by exceeding the maximum Java VM memory size for 32 bit Windows systems (about 2GB).

Classification Tests. For the actual classification tasks, we use the WEKA machine learning tool. The aggregated vectors \vec{a}_f for the different sample files in F are aggregated in a single CSV file to be fed into WEKA. From WEKA's broad range of classification algorithms, we selected the following ones:

- Naïve Bayes
- SMO (a multi-class SVM construct)
- Simple Logistic (regression models)
- J48 (decision tree)
- IB1 (1-nearest neighbor)
- IBk (2-nearest neighbor)

All classifiers are used with their default parameters. The only exception is IBk where the parameter k needs to be set to two to facilitate a 2-nearest neighbor classification. The classifiers work on the following basic principles:

Naïve Bayes is the simplest application of Bayesian probability theory. The SMO algorithm is a way of efficiently solving support vector machines. WEKA's SMO implementation also allows the construction of multi-class classifiers from the two class classifiers intrinsic to support vector machines. Simple Logistic builds linear logistic regression models using LogitBoost. J48 is WEKA's version of a C4.5 decision tree. The IB1 and IBk classifiers, finally, are simple nearest neighbor and k -nearest neighbor algorithms, respectively. All classifiers are applied to the extracted feature vectors created with $n = 256$ and $n = 2048$ samples and various threshold values.

Since only a single set of audio samples is available, all classification tests are performed by splitting this test set. As the splitting strategy we chose 10-fold stratified cross-validation. With this strategy, the sample set is divided into ten subsets of equal size that all contain about the same number of samples from each microphone class (thus the term "stratified"). Each subset is used as the test set in turn, while the remaining nine subsets are combined and used as the training set. This test setup usually gives the most generalizable classification results even for small sample sets. Thus, each of the 10-fold stratified cross-validation tests consists of ten individual classification tasks.

6 Experiments and Results

The classification results for $n = 2048$ are given in Table 3 and are visualized in Figure 2, while the results for the evaluations using 256 frequency coefficients are given in Table 4 and Figure 3. The second column gives percentage of recordings for which the amplitude of at least one window does not exceed the threshold and hence features can be extracted.

Table 3. Classification accuracy for $n = 2048$ (best result for each classifier is highlighted).

Threshold t	Percentage of $w_i(f)$	Naive Bayes	SMO	Simple Logistic	J48	IB1	2-Nearest Neighbor (IBk)
0.01	47.5%	36.3%	54.8%	54.9%	45.5%	53.3%	51.8%
0.025	67.6%	42.9%	66.5%	69.0%	50.6%	67.3%	64.9%
0.05	78.6%	45.7%	76.3%	77.4%	60.4%	74.6%	71.1%
0.1	86.8%	44.6%	79.6%	81.0%	61.5%	79.9%	74.7%
0.25	97.3%	46.9%	88.2%	88.2%	68.6%	82.7%	80.2%
0.35	99.7%	35.7%	90.6%	93.5%	71.6%	88.4%	85.4%
0.40	100.0%	36.5%	88.1%	92.1%	74.0%	88.7%	85.7%
0.5	100.0%	32.3%	83.2%	87.2%	76.8%	88.2%	85.4%
1	100.0%	32.3%	83.2%	87.2%	76.8%	88.2%	85.4%

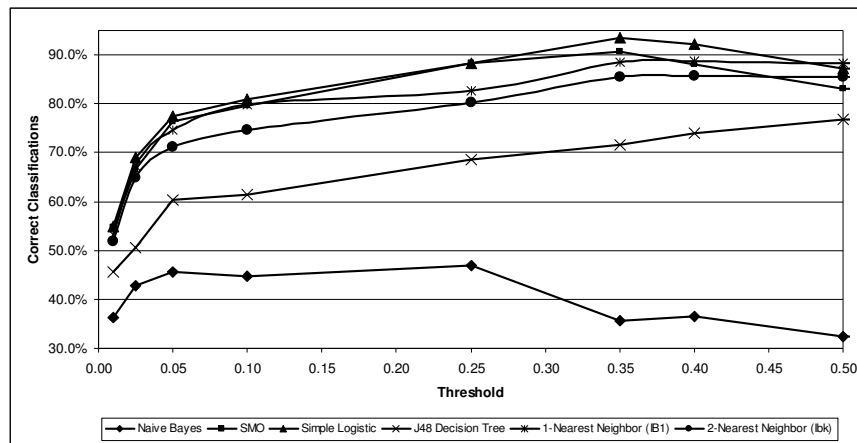


Fig. 2. Graph of the classification accuracy for varying threshold values for $n = 2048$. Results for the threshold of one are omitted since these are identical to those of the threshold of 0.5.

Table 4. Classification accuracy for $n = 256$ (best result for each classifier is highlighted).

Threshold t	Percentage of $w_i(f)$	Naive Bayes	SMO	Simple Logistic	J48	IB1	2-Nearest Neighbor (IBk)
0.01	64.6%	39.1%	52.6%	65.4%	49.3%	60.6%	56.8%
0.025	80.8%	43.3%	63.6%	76.0%	59.1%	74.9%	71.8%
0.05	87.5%	40.2%	63.7%	77.4%	61.0%	74.9%	71.3%
0.1	95.2%	39.4%	72.3%	83.2%	61.9%	75.3%	72.2%
0.2	99.6%	40.5%	74.1%	87.5%	71.7%	83.5%	78.1%
0.225	99.7%	39.3%	74.4%	88.7%	68.2%	83.0%	77.8%
0.25	100.0%	37.6%	74.7%	90.6%	73.4%	87.1%	82.4%
0.5	100.0%	33.0%	70.2%	84.2%	74.4%	86.8%	83.8%

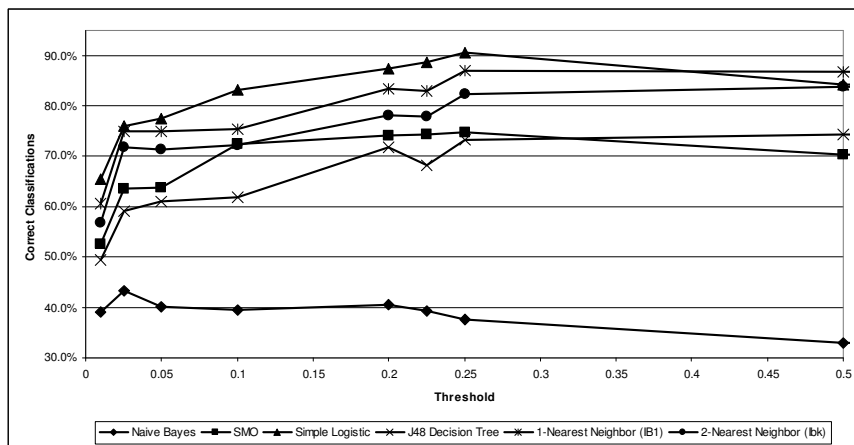


Fig. 3. Graph of the classification accuracy for varying threshold values for $n = 256$. Results for the threshold of one are omitted since these are identical to those of the threshold of 0.5.

The first fact to be observed is that the number of samples for which not a single window falls within the amplitude threshold and which thus can only be classified by guessing is quite high even if the threshold used is set as high as 0.1 of the maximum amplitude – a value at which the audio signal definitely still contains a high portion of audible audio signal in addition to the noise. For all classifiers, the classification accuracy dropped sharply when further reducing the threshold. This result was to be expected since with decreasing threshold, the number of audio samples without any acceptable windows at all increases sharply and thus the classification for more and more samples is based on guessing alone.

For most classifiers, the optimal classification results are obtained with a threshold that is very close to the lowest threshold at which features for all recordings in the test can be extracted (i.e. each recording has at last a single window that lies completely

below the threshold). This, too, is reasonable. For a lower threshold, an increasing number of samples can only be classified by guessing. And for higher thresholds, the amount of signal in the FFT results increases and the amount of noise decreases. Since our classification is based on analyzing the noise spectrum, this leads to lower classification accuracy as well. However, the decline in accuracy even with a threshold of one (i.e. every single sample window is considered) is by far smaller than that of low thresholds.

The classification results for the two window sizes do not differ much. The Naïve Bayes classifier yields better results when using smaller windows and thus fewer attributes. For all other classifiers, the results are usually better for the bigger window size, owing to the fact that a bigger number of attributes allows for the samples to differ in more ways.

The overall best classification results are obtained with the Simple Logistic classifier, with about 93.5% ($n = 2048$) and 90.6% ($n = 256$). However, for very high thresholds that allow a louder audio signal (as opposed to noise) to be part of the extracted features, the IB1 classifier performs better than the Simple Logistic one.

It should be noted that the computation time of the Simple Logistic classifier by far exceeds that of every other classifier. On the test machine (Core2Duo 3 GHz, 4GB RAM), Simple Logistic usually took about 90 minute for a complete 10-fold cross-validation, while the other classifiers only take between a few seconds (Naïve Bayes) and ten minutes.

One notable odd behavior is the fact that for very small thresholds, the percentage of *correctly classified* samples exceeds the percentage of samples with valid windows, i.e. samples that *can be classified*. This is due to the behavior of the classifiers to in essence guess the class for samples without valid attributes. Since this guess is likely to be correct with a probability of one seventh for seven microphones, the mentioned behavior can indeed occur.

Comparison with Earlier Approaches. In [1], a set of 63 segmental features was used. These were based on statistical measures (e.g. entropy, variance, LSB ratio, LSB flip ratio) as well as mel-cepstral features. Their classification results are significantly less accurate than ours, even though they use a test setup based on only four microphones. Their classification accuracy is 69.5% for Naïve Bayes and 36.5% using a k-means clustering approach.

The results in [5] were obtained using the same Fourier coefficient based feature extractor as we did (generating a global feature vector per file instead of segmental features and thereby reducing the complexity of the computational classification task), but its classification was based on a minimum distance classifier. The best reported classification accuracy was 60.26% for $n = 2048$ samples and an amplitude threshold of 0.25. Even for that parameter combination, our best result is an accuracy of 88.2%, while our optimal result is obtained with a threshold of 0.35 (indicating that the new approach is less context sensitive) and has an accuracy of 93.5%.

Principle Component Analysis. In addition to the actual classification tests, a principle component analysis was conducted on the feature vectors for the optimal thresholds (as determined by the classification tests) to determine if the feature space could be reduced and thus the classification be sped up. The analysis uncovered that for $n = 256$, a set of only twelve transformed components is responsible for 95% of the sample variance, while for $n = 2048$, 23 components are necessary to cover the same variance. Thus, the classification could be sped up dramatically without losing much of the classification accuracy.

Inter-Microphone Differences. To analyze the differences in microphone classification accuracy between the individual microphones the detailed classification results for the test case with the most accurate results (Simple Logistic, $n=2048$, threshold $t=0.35$) are shown in a confusion matrix in Table 5.

The results are rather unspectacular. The number of correct classifications varies only slightly, between 89.6% and 96.9% and may not be the result of microphone characteristics, but rather be attributed to differences in recording conditions or to randomness inherent to experiments with a small test set size. The quite similar microphones from the same manufacturer (AKG CK93 and CK98) even get mixed up less often as is the case with other microphone combinations. The only anomaly is the frequent misclassification of the T.Bone SC 600 as the Terratec Headset Master. This may be attributed to these two microphones sharing the same transducer technology, because otherwise, their purpose and price differ considerably.

Table 5. The confusion matrix for the test case Simple Logistic, $n=2048$, $t=0.35$.

Terratec Headset Master	PUX 70 TX-M1	Shure SM 58	T.Bone MB 45	AKG CK 93	AKG CK 98	T.Bone SC 600	classified as
90.60%	1.00%	0.00%	0.00%	0.00%	0.00%	8.40%	Terratec Headset M.
1.00%	95.00%	1.00%	1.00%	1.00%	1.00%	0.00%	PUX 70TX-M1
3.10%	0.00%	89.60%	5.30%	1.00%	0.00%	1.00%	Shure SM 58
0.00%	0.00%	1.00%	97.00%	1.00%	0.00%	1.00%	T.Bone MB 45
1.00%	0.00%	1.00%	2.20%	93.80%	1.00%	1.00%	AKG CK 93
0.00%	0.00%	0.00%	0.00%	1.00%	94.80%	4.20%	AKG CK 98
2.10%	0.00%	2.10%	0.00%	1.00%	1.00%	93.80%	T.Bone SC 600

7 Summary and Future Work

This work showed that it is indeed feasible to determine the microphone model based on an audio recording conducted with that microphone. The classification accuracy can be as high as 93% when the Simple Logistic classifier is used and the features are extracted with 2048 frequency components (4096 samples) per window and the lowest possible threshold that still allows the extraction of features for all samples of the sample set.

Thus, when accuracy is paramount, the Simple Logistic classifier should be used. When computation time is relevant and many attributes and training samples are present, the simple nearest neighbor classifier represents a good tradeoff between speed and accuracy.

As detailed in the introduction, these results do by no means represent a definite answer in finding the optimal technique for microphone classification. They do, however, demonstrate the feasibility of such an endeavor. The research conducted for this project also led to ideas for future improvements:

In some cases, the audio signal recorded by the microphone is a common one and an original version of it could be obtained. In these cases, it could be feasible to subtract the original signal from the recorded one. This may lead to a result that contains only the distortions and noise introduced by the microphone and may lead to a much more relevant feature extraction.

For our feature extractor, we decided to use non-overlapping FFT windows to prevent redundancy in the data. However, it might be useful to have the FFT windows overlap to some degree, as with more sample windows, the effect of randomness on the frequency histogram can be reduced. This is especially true if a high number of windows is being rejected by the thresholding decision.

Another set of experiments should be conducted to answer the question on whether our classification approach can also be used for microphone identification, i.e. to differentiate even between different microphones of the same model.

Furthermore, additional features could be used to increase the discriminatory power of the feature vector. Some of these were already mentioned in [1] and [5], but were not yet used in combination. These features include – among others – the microphone's characteristic response to a true impulse signal and the dynamic range of the microphone as analyzed by recording a sinus sweep.

Finally, classifier fusion and boosting may be valuable tools to increase the classification accuracy without having to introduce additional features.

Acknowledgments

We would like to thank two of our students, Antonina Terzieva and Vasil Vasilev who conducted some preliminary experiments leading to this paper and Marcel Dohnal for the work he put into the feature extractor. The work in this paper is supported in part by the European Commission through the FP7 ICT Programme under Contract FP7-ICT-216736 SHAMAN. The information in this document is provided as is, and no guarantee or warranty is given or implied that the information is fit for any particular purpose. The user thereof uses the information at its sole risk and liability.

References

1. Kraetzer, C., Oermann, A., Dittmann, J., and Lang, A.: Digital Audio Forensics: A First Practical Evaluation on Microphone and Environment Classification. In: 9th Workshop on Multimedia & Security, pp. 63--74. ACM, New York (2007)
2. Witten, I.H., Frank, E.: Data Mining: Practical Machine Learning Tools and Techniques, 2nd Edition, Morgan Kaufmann, San Francisco (2005).
3. SHAMAN - Sustaining Heritage Access through Multivalent ArchiviNg, <http://shaman-ip.eu>
4. SoX – Sound Exchange, <http://sox.sourceforge.net>
5. Donahl, M.: Forensische Analyse von Audiosignalen zur Mikrofonerkennung, Masters Thesis, Dept. of Computer Science, Otto-von-Guericke University Magdeburg, Germany, (2008).
6. Filler, T., Fridrich, J., Goljan, M.: Using Sensor Pattern Noise for Camera Model Identification, In Proc. ICIP08, pp. 1296-1299, San Diego (2008)
7. Dirik, A.E., Sencar, H.T., Memon N.: Digital Single Lens Reflex Camera Identification From Traces of Sensor Dust. IEEE Transactions on Information Forensics and Security Vol. 3, 539--552 (2008).
8. Gloe, T., Franz, E., Winkler A.: Forensics for flatbed scanners. In: Proceedings of the SPIE International Conference on Security, Steganography, and Watermarking of Multimedia Contents, San Jose (2007)
9. Khanna, N., Mikkilineni, A.K., Chiu, G.T., Allebach, J.P., Delp, E.J.: Survey of Scanner and Printer Forensics at Purdue University. In: IWCF 2008. LNCS, vol. 5158, pp. 22--34. Springer, Heidelberg (2008)
10. Bayram, S., Sencar, H.T., Memon, N.: Classification of Digital Camera-Models Based on Demosaicing Artifacts. Digital Investigation, vol. 5, issues 1-2, 49--59 (2008)
11. Oermann, A., Vielhauer, C., Dittmann, J.: Sensometrics: Identifying Pen Digitizers by Statistical Multimedia Signal Processing. In: SPIE Multimedia on Mobile Devices 2007, San Jose (2007)
12. Maher, R.C.: Acoustical Characterization of Gunshots. SAFE 2007 11-13 April 2007, Washington D.C., USA (2007).
13. Oermann, A., Lang, A., Dittmann, J.: Verifier-tuple for audio-forensic to determine speaker environment. In: Proc. MM & Sec'05, pp. 57 - 62, New York (2005)