Spectral Denoising for Microphone Classification

Luca Cuccovillo
Fraunhofer IDMT
Ilmenau, Germany
luca.cuccovillo@idmt.fraunhofer.de

Antonio Giganti
Politecnico di Milano
Milano, Italy
antonio.giganti@polimi.it

Paolo Bestagini
Politecnico di Milano
Milano, Italy
paolo.bestagini@polimi.it

Patrick Aichroth
Fraunhofer IDMT
Ilmenau, Germany
patrick.aichroth@idmt.fraunhofer.de

Stefano Tubaro
Politecnico di Milano
Milano, Italy
stefano.tubaro@polimi.it

ABSTRACT

In this paper, we propose the use of denoising for microphone classification, to enable its usage for several key application domains that involve noisy conditions. We describe the proposed analysis pipeline and the baseline algorithm for microphone classification, and discuss various denoising approaches which can be applied to it within the time or spectral domain; finally, we determine the best-performing denoising procedure, and evaluate the performance of the overall, integrated approach with several SNR levels of additive input noise. As a result, the proposed method achieves an average accuracy increase of about 25% on denoised content over the reference baseline.

CCS CONCEPTS

• **Security and privacy**; • **Information systems** → *Multimedia information systems*; *Social networks*; • **Computing methodologies** → **Artificial intelligence**; *Machine learning*;

KEYWORDS

Microphone classification, Audio forensics, Spectral denoising, AI-based denoising, Digital signal processing, Machine learning

1 INTRODUCTION

The recent surge of disinformation, which often includes malevolent manipulation, decontextualization or fabrication of audio-visual material via social media, has drawn increasing attention from the research community, which reacted by developing innovative algorithms for detecting frauds and analysing controversial multimedia content [1, 17]. In particular, the discipline of multimedia forensics has found a new momentum: The need for tools to analyze acquisition and processing traces within content, and (i) to compare them

with alleged information about the content [5, 16] and (ii) to use it to detect and localize manipulations [2, 6]. Such tools are crucial to fight media disinformation, both in investigative journalism, and in courtroom cases.

In this paper, we address an important challenge related to microphone classification, a classic task within the forensics domain that aims at identifying which device has been used to record a given audio item [5, 6, 13–15]: Until now, microphone classification algorithms have focused on the analysis of fairly high-quality content, which is common e.g. in courtroom cases. They are therefore very sensitive to background or additive noise. This sensitivity to noise, however, is greatly reducing the applicability of microphone classification to the kind of audio-visual disinformation that is increasingly shared via social media, which often includes noisy audio material.

Considering this, the motivation for this paper was to investigate whether and which denoising techniques based either on Digital Signal Processing (DSP) or Artificial Intelligence (AI) could be applied as a pre-processing step for microphone classification, to improve its robustness against noise. Moreover, we propose an integrated algorithm for microphone classification using such denoising: It is is based on our own pre-existing baseline for closed-set microphone classification [6], later extended also to address an open-set setup [5], and uses the best-performing denoising algorithm we were able to identify within this work: the AI-based DnCNN architecture for image denoising proposed by Zhang et al. [21], which we applied to the spectral domain. This new application of AI to spectral denoising for microphone classification achieved very promising results, with an average accuracy increase of about 25% in comparison to the baseline.

The further chapters of the paper are organized as follows: In Section 2, we present our proposed integrated approach for microphone classification, based on the baseline algorithm that is described in Section 3. In Section 4, we present various denoising techniques, which are then compared in Section 5 to select the most suitable one for microphone classification. Finally, in Section 6, we evaluate the proposed integrated approach with varying levels of additive noise. Section 7 closes with a a discussion about future research directions and ideas for further improvements.

2 PROPOSED APPROACH

Our integrated approach for microphone classification consists of three main components:

(1) A *denoising* component, which analyses a noisy input signal $\tilde{x}$ and returns an estimate $\hat{x}$ of the input signal x before being affected by any additive noise

(2) A *log-power extraction* component, which takes an input signal in the time domain $x(t)$ and returns its logarithmic power spectrum $X(f)$

(3) A *microphone classification* component, which uses an input power spectrum $X(f)$ to return the *label* of the microphone device used for recording the initial audio signal

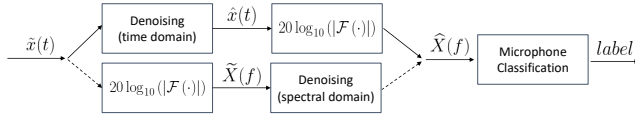As depicted in Figure 1, the order of the described steps depends on the target domain of the denoising component.

Figure 1: Overview of the integrated approach

If the denoising component is designed to work on a signal $x(t)$ in the time domain, then denoising is applied first, to be followed by log-power extraction. The denoised log-power is then used by microphone classification. This process is described more formally with eqs. (1a) to (1c)

$$\hat{x}(t) = \text{denoise}\left(\tilde{x}(t)\right) \approx x(t), \tag{1a}$$

$$\widehat{X}(f) = 20 \log_{10}\left(|\mathcal{F}(x(t))|\right), \tag{1b}$$

$$label = \text{classify}\left(\widehat{X}(f)\right), \tag{1c}$$

where $\mathcal{F}(\cdot)$ denotes the Fourier transform, and $x(t)$ the original signal in the time domain before the noise addition.

If the denoising component is designed to work on a signal $X(f)$ in the frequency domain, then log-power extraction is performed first, followed by denoising that is applied as desired on the frequency domain. Again, the denoised log-power is then used by microphone classification. This process is described more formally with eqs. (2a) to (2c)

$$\widetilde{X}(f) = 20 \log_{10}\left(|\mathcal{F}(\tilde{x}(t))|\right), \tag{2a}$$

$$\widehat{X}(f) = \text{denoise}\left(\widetilde{X}(f)\right) \approx X(f), \tag{2b}$$

$$label = \text{classify}\left(\widehat{X}(f)\right), \tag{2c}$$

where $X(f)$ denotes the original signal in the frequency domain before the noise addition.

The denoising components in the frequency domain that can be applied to the integrated approach are less constrained than "classic" ones: Since we are not going to convert the denoised signal $\widehat{X}(f)$ back into the time domain, candidate algorithms can ignore the phase of the signal completely, focusing only on its log-power. Moreover, we do not need to work on the whole signal length at once, but can rather focus on a set of L analysis frames, each of which being denoted by $x_l(t)$. An important consequence of these relaxed constraints is that if we consider the whole Short Time Fourier Transform (STFT) of the signal at once, i.e., a set of L frames in which the log-power of the l-th frame can be denoted by $X_l(f)$, the denoising operation can be performed also by means of *image*

denoising algorithms, which have been thoroughly investigated [3, 4, 19–21].

3 MICROPHONE CLASSIFICATION BASELINE

The baseline approach for microphone classification consists of our own previous method that is based on blind channel estimation [5, 6].

3.1 Channel Estimation

The algorithm starts from the assumption that each frame $x_l(t)$ of an input recording $x(t)$ can be modeled by a convolution between a *fixed* transmission channel $h(t)$, and the original input speech $s_l(t)$, i.e.:

$$x_l(t) = h(t) * s_l(t), \tag{3}$$

where we assume the transmission channel equal to the *frequency response* of the recording device.

An equivalent formulation in the log-power domain is:

$$X_l(f) = H(f) + S_l(f), \tag{4}$$

which provides us with a straight-forward solution to estimate the log-power $H(f)$ of the transmission channel: If we can estimate the ideal input speech $S_l(f)$, i.e., if we can compute a term $\widehat{S}_l(f)$ which is accurate enough, the channel can be estimated blindly by applying:

$$\widehat{H}(f) = \frac{1}{L} \sum_{l=1}^{L} \left(X_l(f) - \widehat{S}_l(f) \right), \tag{5}$$

with L denoting the amount of frames and thus eq. (5) denoting the average difference between the input recording frames and the estimated ideal speech.

To further improve the stability of eq. (5), we can normalize both terms $X_l(f)$ and $\widehat{S}_l(f)$ to have zero mean, by defining their normalized equivalents

$$Z_{X_l}(f) = X_l(f) - \frac{1}{N_{\text{stft}}} \sum_{f=1}^{N_{\text{stft}}} X_l(f), \tag{6a}$$

$$Z_{\widehat{S}_l}(f) = \widehat{S}_l(f) - \frac{1}{N_{\text{stft}}} \sum_{f=1}^{N_{\text{stft}}} \widehat{S}_l(f), \tag{6b}$$

and obtain the final eq. (7) for the mean-normalized blind estimation of the microphone frequency response:

$$\hat{h} = \frac{1}{L} \sum_{l=1}^{L} \left(Z_{X_l}(f) - Z_{\widehat{S}_l}(f) \right). \tag{7}$$

3.2 Ideal Speech Estimation

The ideal speech estimate $\widehat{S}_l(f)$ in eq. (5) can be retrieved by leveraging spectrum classification, as firstly proposed by Gaubitch et al. [9].

The first step for of the procedure consists of processing a large speech corpus to extract a high amount of RASTA filtered Mel-frequency cepstral coefficients (MFCCs) [11], i.e., MFCCs which are more robust to channel effects than the usual formulation, and are suited to represent phonemes. In the following, MFCCs of the l-th frame of an input audio signal x will be denoted by the symbol c_{X_l}.

Given L_X training MFCC vectors c_{X_l}, used to fit a Gaussian Mixture Model (GMM) with M mixtures, a key element of the

estimation procedure is the relative mixture probability $p_i\left(c_{X_l}\right)$, i.e., the probability that the feature vector c_{X_l} belongs to the i-th mixture:

$$p_i\left(c_{X_l}\right) = \frac{\pi_i \cdot \mathcal{N}\left(c_{X_l} \mid \mu_i, \Sigma_i\right)}{\sum_{m=1}^{M} \pi_m \cdot \mathcal{N}\left(c_{X_l} \mid \mu_m, \Sigma_m\right)}. \tag{8}$$

In eq. (8), $\mathcal{N}\left(c_{X_l} \mid \mu_i, \Sigma_i\right)$ denotes the posterior probability of the vector c_{X_l} against the i-th mixture, having a normal distribution with mean μ_i, diagonal covariance Σ_i, and prior π_i.

Given these definitions, a model of the average log spectrum of the ideal speech can be obtained as follows:

(1) Build a first normalized power spectrum matrix $\overline{Z}_X$, by collecting all mean-normalized log powers $Z_{X_l}(f)$ of the GMM training set:

$$\overline{Z}_X \in \mathbb{R}^{L_X \times N_{\text{stft}}} = \left\{Z_{X_l}(f)\right\}, \tag{9a}$$

(2) Build a relative probability matrix $\overline{P}_X$, by collecting all relative mixture probabilities $p_i\left(c_{X_l}\right)$ of the GMM training set:

$$\overline{P}_X \in \mathbb{R}^{L_X \times M} = \left\{p_i\left(c_{X_l}\right)\right\} \tag{9b}$$

(3) Compute the average speech spectrum matrix $\overline{S}_X$:

$$\overline{S}_X \in \mathbb{R}^{M \times N_{\text{stft}}} = \overline{P}_X^t \cdot \overline{Z}_X, \tag{9c}$$

with t denoting the transposition.

$\overline{S}_X$, also depicted in Figure 2, is at the core of the ideal speech estimation procedure: Given an arbitrary input speech signal s having L_S frames and a relative probability matrix $\overline{P}_S$, it is straightforward to compute:

$$\overline{Z}_{\widehat{S}} \in \mathbb{R}^{L_S \times N_{\text{stft}}} = \overline{P}_S \cdot \overline{S}_X, \tag{10}$$

i.e., a matrix the columns of which can be directly applied in eq. (7), to obtain the desired estimate of the microphone frequency response.

If we analyse eq. (10) in more detail, we can observe that the relative probability matrix $\overline{P}_S$ acts as a selection matrix for the rows of the average speech spectrum $\overline{S}_X$. In other words, the ideal speech estimate $\widehat{S}_l(f)$ is obtained by means of a convex combination of the rows of $\overline{S}_X$. The average speech spectrum matrix can thus be interpreted as a dictionary of phonemes, which can be composed for producing any arbitrary input speech signal.

3.3 Closed-Set Classification

The last step of the baseline consists of a feature vector computation, and the actual training of the classifier. For the sake of reproducibility, we use the channel estimate $\hat{h}$ in eq. (10) as feature vector for the classification. Moreover, we use a classic Support Vector Machine (SVM) with a Radial Basis Function (RBF) kernel to perform closed-set classification, and refrain from addressing open-set classification[1].

---

[1]The extension of the baseline in [6] to an open-set scenario was addressed in [5], and is compatible with this new proposal to apply denoising as pre-processing step.
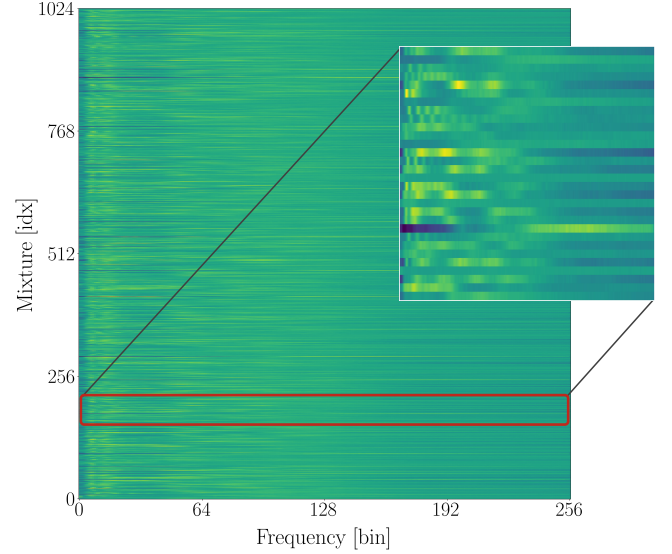

Figure 2: Average Speech Spectrum Matrix $\overline{S}_X$

4 DENOISING BASELINES

In the literature, a broad variety of denoising techniques have been proposed to address the denoising problem on signals of different nature for various kinds and levels of noise. In our work, we focus on a selection of well-established DSP-based and more modern AI-based solutions that can be applied to either the time-variant audio signal, or to its time-frequency representation considered as a 2D image.

4.1 DSP-based: Total Variation

Image denoising based on total variation has been proposed in [19]. This method is based on the minimization of a constrained optimization problem. Specifically, the method aims at finding the image with minimum total variation that minimizes the mean square error with respect to the noisy observation.

Let us define a noisy image as

$$v = u + n, \tag{11a}$$

where u is the ideal noise-free image, and n is an additive noise term. By denoting the pixel in position ij of image u as u_{ij}, we can define the total variation $V(u)$ of the clean image u as the L_1 norm of its gradient

$$V(u) = \sum_{i,j} \sqrt{\left|u_{i+1,j} - u_{ij}\right|^2 + \left|u_{i,j+1} - u_{ij}\right|^2}, \tag{11b}$$

where $|\cdot|$ denotes the absolute value.

Given a noisy image v, image denoising based on total variation consists in estimating the clean image $\hat{u}$ by solving a minimization problem defined as

$$\hat{u} = \arg\min_{u} \sum_{i,j} \sqrt{\left|u_{ij} - v_{ij}\right|^2} + \lambda V(u), \tag{11c}$$

where λ is a regularization parameter to weight the fidelity term (first term of the equation representing the mean squared error

between u and v) and the total variation of the estimated u (second term of the equation).

4.2 DSP-based: Non-Local Means

Denoising based on non-local means is an image denoising algorithm proposed in [3], which performs the denoising by replacing each pixel by a weighted average of all *similar* pixels in the rest of the image.

Given a noisy picture v, the non-local means algorithm retrieves the corresponding clean picture $\hat{u}$ by applying

$$\hat{u}(i) = \frac{1}{C(i)} \sum_{j} w(i, j) v(j), \tag{12a}$$

where $\hat{u}(i)$ is the i-th pixel of image $\hat{u}$, $v(j)$ represents the j-th pixel of image v, $C(i)$ acts as a normalisation term defined as

$$C(i) = \sum_{j} w(i, j), \tag{12b}$$

and the weighting factor $w(i, j)$ depends on the similarity between pixels of v in a neighborhood of its i-th pixel and pixels of v in a neighborhood of its j-th pixel.

4.3 DSP-based: Bilateral Filtering

Bilateral filtering is an image denoising algorithm proposed in [20], in which each pixel is replaced by a weighted average of *similar, nearby* pixels.

Given a neighborhood $\mathcal{N}_j$ of the j-th pixel coordinates of a noisy image v, bilateral filtering can be used to recover an estimate of the corresponding clean picture u by applying:

$$\hat{u}(i) = \frac{1}{W_p} \int_{j \in \mathcal{N}_j} v(j) \cdot s\left(v(j), v(i)\right) \cdot c\left(j, i\right) \, dj, \tag{13a}$$

where $s(\cdot)$ is a *similarity* function determining how much the pixel values are alike, $c(\cdot)$ a *closeness* function determining how much the pixel coordinates are close to each other, and W_p a normalization factor.

The authors suggested to use Gaussian functions of the Euclidean distances to determine both similarity and closeness:

$$s\left(v(j), v(i)\right) = \exp \frac{1}{2} \left(\frac{\|v(j) - v(i)\|}{\sigma_s}\right)^2, \tag{13b}$$

and

$$c\left(j, i\right) = \exp \frac{1}{2} \left(\frac{\|j - i\|}{\sigma_c}\right)^2, \tag{13c}$$

with σ_s determining the similarity (photometric) spread, and σ_c being the closeness (geometric) spread.

4.4 DSP-based: Wavelet BayesShrink

The wavelet BayesShrink is an image denoising algorithm proposed in [4], which leverages wavelet decomposition to filter-out high frequency components associated to the noise.

The authors proposed to focus on the *detail* sub-bands HH_k, where k is the scale, computed by applying the two-dimensional dyadic orthogonal wavelet transform operator $\mathcal{W}$ to the noisy image v, yielding the coefficient coefficient matrix

$$V = \mathcal{W}v. \tag{14a}$$

In particular, the authors proposed to perform a soft-threshold function to all coefficients V_{ij} of the detail sub-bands, applying

$$\hat{U}_{ij} = \begin{cases} \text{sgn}(V_{ij}) \cdot \max\left(\left|V_{ij}\right| - T, 0\right), & \text{if } V_{ij} \in HH_k \\ V_{ij}, & \text{else} \end{cases}, \tag{14b}$$

and then transform the image back, thus leaving all low-resolution coefficients unaltered:

$$\hat{u} = \mathcal{W}^{-1}\hat{U}. \tag{14c}$$

The core contribution of the paper consisted in defining an *optimal* data-driven threshold $T^{\star}$ for the shrinking equation in (14b), which minimized the Bayesian risk function associated to the denoising:

$$T^{\star} = \arg\min_{T} E(\hat{U} - U)^2. \tag{14d}$$

The optimal threshold must be computed per each detailed-subband, and is equal to

$$T^{\star} = \frac{\hat{\sigma}^2}{\hat{\sigma}_U}, \tag{14e}$$

where

$$\hat{\sigma} = \frac{\text{Median}\left(\left|Y_{ij}\right|\right)}{0.6745}, \forall Y_{ij} \in HH_1 \tag{14f}$$

and

$$\hat{\sigma}_U = \sqrt{\max\left(\frac{1}{n^2} \sum_{i,j=1}^{n} V_{ij}^2, 0\right)}, \tag{14g}$$

with $n \times n$ being the size of the k-th detailed subband.

4.5 AI-based: Denoising CNN Architecture

The Denoising CNN architecture (DnCNN) in [21] is a AI-based approach for image denoising aiming at solving the problem by correctly predicting not the original clean image u, but rather the *residual* noise n.

The authors start by defining the network as a function $\mathcal{F}(\cdot)$ with parameters Θ, which takes a noisy image $v = u + n$ as input, and performs the following mapping:

$$\mathcal{F}(v, \Theta) \approx n, \tag{15a}$$

with n being equal to the (Gaussian white) noise to be removed.

The loss function $l(\cdot \mid \Theta)$ to be minimized can thus be defined as the mean squared error between the target noise and the one estimated from the noisy input:

$$l(v, u \mid \Theta) = \frac{1}{N} \sum_{i=1}^{N} \|\mathcal{F}(v(i), \Theta) - (v(i) - u(i))\|^2, \tag{15b}$$

with N being the amount of pixels in the signal. The advantage of such a procedure, is that a large amount of noise examples can be generated online during training, and thus the network can learn how to predict noises at several Signal-to-Noise Ratio (SNR) levels – and potentially more diverse than uniform or Gaussian white noise.

4.6 AI-based: Audio Denosing Autoencoder

The Audio Denoising Autoencoder (Audio DAE) in [7] is a AI-based approach for audio denoising in the time domain, aiming at reconstructing the clean audio signal $u(t)$ from its noise corrupted version $v(t) = u(t) + n(t)$.

The architecture acts as a denoising autoencoding function $\mathcal{F}(\cdot)$ with parameters Θ, thus realizing the following mapping:

$$\mathcal{F}(v(t), \Theta) \approx u(t). \tag{16}$$

In order to train the network the authors defined the STFT loss $l_{stft}(\Theta)$ as the composition of a *spectral convergence* loss $l_{sc}(\Theta)$ and a *magnitude* loss $l_{mag}(\Theta)$:

$$l_{stft}(u, \hat{u} \mid \Theta) = l_{sc}(u, \hat{u} \mid \Theta) + l_{mag}(u, \hat{u} \mid \Theta), \tag{17}$$

where

$$l_{sc}(u, \hat{u} \mid \Theta) = \frac{\||STFT(u)| - |STFT(\hat{u})|\|_F}{\||STFT(u)|\|_F}, \tag{18}$$

$$l_{mag}(u, \hat{u} \mid \Theta) = \frac{1}{T}\|\log|STFT(u)| - \log|STFT(\hat{u})|\|_1, \tag{19}$$

with T denoting the input length, $\|\cdot\|_1$ the L_1 norm and $\|\cdot\|_F$ the Frobenius norm.

The peculiarity of the STFT loss is that it can be computed with several configurations, using different number of STFT bins, hop sizes and window lengths. Given M such configurations, the authors finally determined the required training loss for the whole architecture:

$$l(u, \hat{u} \mid \Theta) = \frac{1}{T}\left(\|u - \hat{u}\|_1 + \sum_{m=1}^{M} l_{stft}^{(m)}(u, \hat{u} \mid \Theta)\right). \tag{20}$$

This composite loss is meant to ensure phase and magnitude coherency of the output audio signal, while at the same time avoiding artifacts which would have appeared by using only one STFT resolution.

5 DENOISER SELECTION

In order to determine if and to which extent the denoising baselines outlined in Section 4 are applicable for microphone classification, we compared their performances on audio log-powers, which were extracted from the MOBIPHONE [12] dataset according to the overview in Figure 1.

The MOBIPHONE dataset was collected by recording 10 utterances from 24 speakers, using 21 mobile phones of various models from 7 different brands. After its publication it became a common dataset for benchmarking microphone classification algorithms, and we thus decided to use it both for comparing the denoiser performances, and for running the complete system evaluation in Section 6.

The dataset preparation for the denoiser comparisons was performed as follows:

(1) We corrupted all audio files in the MOBIPHONE dataset with additive white Gaussian noise, using a SNR of 25 dB.
(2) We extracted log-power spectrograms from the clean MOBIPHONE dataset, obtaining a reference set $X_{ideal}(f)$
(3) We extracted denoised log-power spectrograms from the noisy audio files, obtaining a benchmark set $\widehat{X}_{25}(f)$

(4) We split both reference set and benchmark set into training and testing portions, obtaining the four distinct sets $X_{ideal}^{train}(f), \widehat{X}_{25}^{train}(f), X_{ideal}^{test}(f), \widehat{X}_{25}^{test}(f)$

After obtaining the four aforementioned sets, we compared the outcome of the denoising using 3 different metrics:

(1) PSNR: Average Peak Signal-to-Noise Ratio (PSNR) between corresponding pairs of $X_{ideal}^{test}(f)$ and $\widehat{X}_{25}^{test}(f)$
(2) SSIM: Average Structural Similarity Index Measure (SSIM) between corresponding pairs of $X_{ideal}^{test}(f)$ and $\widehat{X}_{25}^{test}(f)$
(3) MCA: Microphone Classification Accuracy (MCA) of the baseline trained on $X_{ideal}^{train}(f)$ and tested on $\widehat{X}_{25}^{test}(f)$

The first two metrics relate directly to the visual quality of the denoising: The PSNR quantifies the closeness in terms of pixel energy between the original spectrogram and the denoised one, while the SSIM their similarity in terms of luminance, contrast and structure. The MCA metric is meant to capture to which extent the classification is possible after the denoising operation: Aggressive denoisers may remove too much content from the log-power spectrogram, while ineffective ones may remove too little disturbance for being of any help.

The outcome of this evaluation is reported in Table 1, with example denoised spectrograms being depicted in Figure 3.

Table 1: Denoising Baseline Benchmarks

Denoiser	Performances (#)		
	PSNR	SSIM	MCA
Total Variation [19]	20.56	0.81	34.40
Non-Local Means [3]	20.59	0.83	36.69
Bilater Filtering [20]	20.33	0.81	33.67
Wavelet BayesShrink [4]	20.62	0.82	39.06
DnCNN Architecture [21]	**27.80**	**0.86**	**69.09**
Audio DAE Architecture [7]	21.02	0.76	26.47

We can observe that the four DSP-based denoising algorithms perform relatively on-par in terms of both visual quality and microphone accuracy. The latter, however, is not sufficient for practical applications, given the MCA score below 40%.

The Audio DAE is superior to DSP-based denoisers in terms of PSNR, but achieves a lower SSIM. The reason for this counter-intuitive behavior can be identified by checking Figure 3: The Audio DAE is pretty aggressive within the silent portions of the audio signal, and generates some edges in the spectrogram that lower the SSIM. The same aggressiveness is probably also the cause for a further decrease of the MCA score, which is reduced to about 26%.

The DnCNN Architecture seems to be the most promising alternative: Its PSNR is superior to the Audio DAE by more than 6dB, and the SSIM is beyond the one achieved by the DSP-base denoisers. If we consider the examples in Figure 3, the higher SSIM is probably due to the speech components being sharper than in other spectrograms. The most promising score, however, is the MCA itself: Without retraining the classifier, which would probably improve
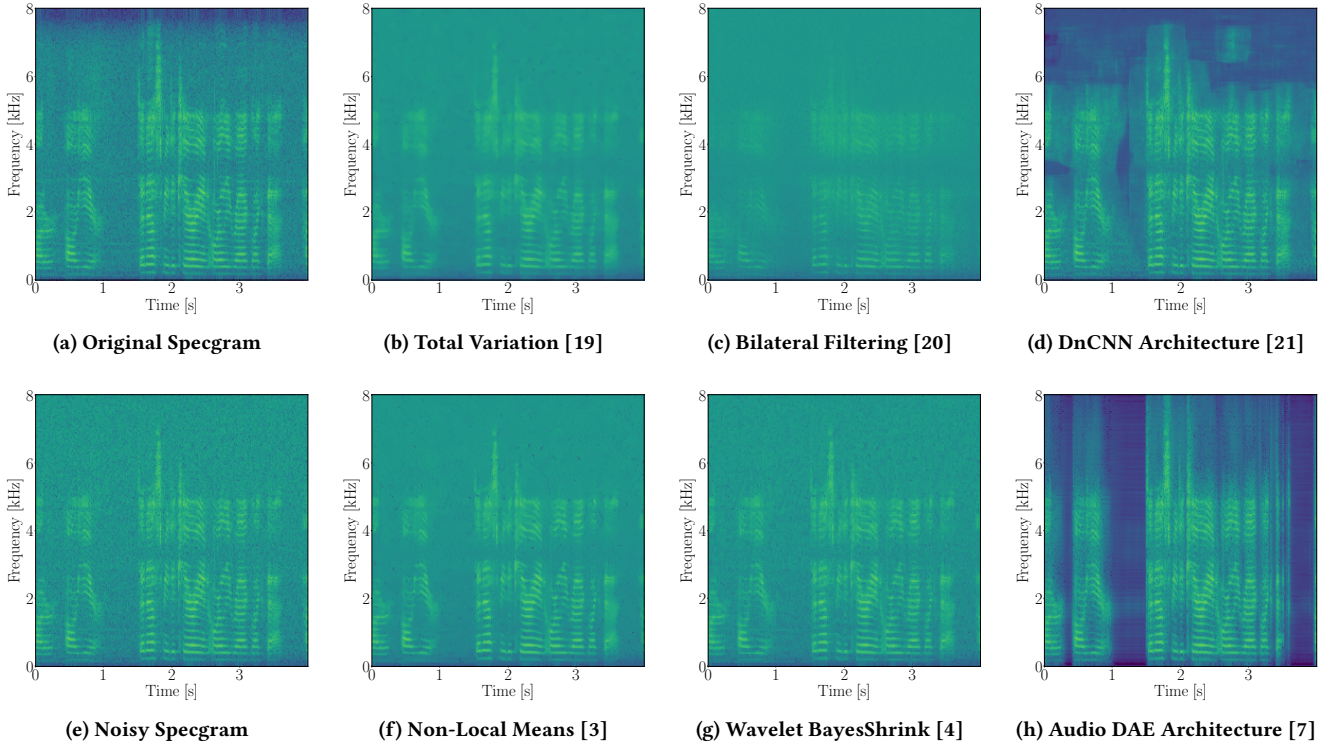
(a) Original Specgram   **(b) Total Variation [19]**   **(c) Bilateral Filtering [20]**   **(d) DnCNN Architecture [21]**

(e) Noisy Specgram   **(f) Non-Local Means [3]**   **(g) Wavelet BayesShrink [4]**   **(h) Audio DAE Architecture [7]**

Figure 3: Example denoising of log-power spectrograms, with audio SNR=25dB

the performances but could be a costly operation, we achieved an accuracy of about 69%, which is significantly beyond the accuracy of all other alternatives. We hence selected the DnCNN Architecture to perform the complete system evaluation, which is outlined in the next section.

6 EVALUATION

The final evaluation of our proposed integrated approach, combining AI-based denoising performed by a DnCNN architecture [21] and microphone classification based on blind channel estimation [5, 6], involved several datasets in conjunction.

Similarly as for the denoiser selection, the core evaluation dataset was the MOBIPHONE dataset, which was resampled to 16 kHz and split in segments of 4.096 seconds. This segment length, in conjunction with a STFT using 512 points and 50% overlap, led us to a set of 180 log-power spectrogram examples of size 256×256 per each of the 21 classes in the dataset. Furthermore, we split training and testing examples according to a speaker-wise logic: Spectrograms related to the utterances of the first 19 speakers were used for training (142 examples per class), and the remaining ones for testing (38 examples per class).

In addition to the original MOBIPHONE dataset, we created noise-corrupted versions of both the train and the test examples, using audio SNRs equals to 20,35,30 and 35 dB. If we assume that the initial recordings had infinite SNR, each set (and its denoised

equivalent) can be uniquely identified as follows:

X_{snr}^{train} : Noisy (SNR=snr) training examples

X_{snr}^{test} : Noisy (SNR=snr) test examples

$\widehat{X}_{snr}^{train}$: Denoised (from SNR=snr) training examples

$\widehat{X}_{snr}^{test}$: Denoised (from SNR=snr) test examples

The SVM used for microphone classification was trained *once* on the X_{∞}^{train} set, selecting a RBF kernel with $\gamma = 1/(256\,\sigma_x^2)$ –where σ_x^2 is equal to the variance of the normalized feature vectors–, and then tested separately on each $\widehat{X}_{snr}^{test}$. The absence of retraining has the advantage of simulating the behavior of *pre-existing* classification pipelines which are exposed to noisy content. The DnCNN used for spectrogram denoising was instead trained on each available $\left(X_{snr}^{train}, X_{\infty}^{train}\right)$ pair. Due to to the speaker-wise split, test examples where thus unseen for both the denoiser and the classifier.

The GMM included in the microphone classification baseline was trained by combining utterances from the LibriSpeech corpus [18], until reaching a total duration of one hour. According to [6], we trained a GMM with 1024 mixtures, using 12 MFCCs per frame, computed for each frame of the aforementioned denoised STFT. Being the speakers in the LibriSpeech corpus absent from the MOBIPHONE dataset – which includes utterances from TIMIT [8] – we tried to ensure, once again, that examples were completely unknown to the integrated classification algorithm.

The final outcome of the evaluation is presented in Table 2 and Table 3. The results in Table 2 describe the performance of the system if applied without any precautions to noisy content: From

being nearly perfect – with accuracy, precision and recall all higher than 99% – all metrics drop by more than half if the quality decrease to a SNR of 30 dB of lower. Even a very light noise with 35 dB SNR is enough to lower the accuracy to about 60%. The loss of performances was not unexpected, due to the lack of any noise term in the channel estimation formulation of eq. (7), but is still higher than we would have imagined, and confirms the need for countermeasures whenever analysing noisy input audio files.

Table 2: Results *without* including denoising

SNR (dB)	Performances (%)		
	Accuracy	Precision	Recall
+∞	99.21	0.992	0.992
35	60.65	0.643	0.603
30	50.36	0.514	0.500
25	41.81	0.437	0.415
20	36.05	0.409	0.353

The results in Table 3, instead, describe the performance of the system we proposed, in which the denoising is active and the classifier is not retrained nor aware of the denoising taking place. We see a very consistent behavior in performances for all noise-corrupted cases: Accuracy, precision and recall, which dropped severely without countermeasures, raise by about 25% compared to the baseline, and at least for low-noise scenarios – SNR of at least 30 dB – they consistently score beyond 80%. There is a perceivable decrease compared to the initial 99% of accuracy, but the system can still provide useful results. Analysis performances for clean content decrease slightly to about 95%, i.e., the denoising procedure is not completely transparent, but not to a prohibitive point. We will discuss a mitigation strategy for this issue and other further research directions in the following section.

Table 3: Results of our proposed approach

SNR (dB)	Performances (%)		
	Accuracy	Precision	Recall
+∞	95.11	0.955	0.950
35	84.17	0.872	0.841
30	83.71	0.861	0.839
25	69.09	0.771	0.694
20	57.54	0.695	0.578

7 CONCLUSIONS AND OUTLOOK

To our knowledge, this work represents the first systematic approach addressing the influence of noise on microphone classification. The proposed pipeline was tested on a widely accepted benchmark dataset, and proved to be effective in lessening the negative influence of noise on the classification task: The proposed algorithm achieved an accuracy beyond 80% for noisy conditions with audio SNR of at least 30dB, providing an accuracy increase of

about 25% in comparison with the initial pipeline, *without* the need for re-training the classifier under analysis.

In a parallel submission [10], we will address two additional research questions which we could not cover within this pages: Testing whether DnCNN-based denoising can be applied successfully to other state-of-the-art feature vectors for microphone classification, and investigating whether data augmentation can boost the overall accuracy and avoid the slight performance drops for clean-yet-denoised test files.

Furthermore, we plan to verify the efficacy of the DnCNN denoising with alternative classification algorithms, and investigate countermeasures to more complex noises than the uniform white Gaussian noise addressed by this publication, e.g., babble noise, car noise or background music.

We will also investigate whether a similar approach could be used to lessen the negative influence of lossy compression and transcoding, which are also likely to occur on social networks. Finally, we plan to experiment with new denoising techniques which are likely to emerge within the image-processing domain.

ACKNOWLEDGMENTS

REFERENCES

[1] Christina Boididou, Symeon Papadopoulos, Lazaros Apostolidis, and Yiannis Kompatsiaris. 2017. Learning to Detect Misleading Content on Twitter. In *ACM International Conference on Multimedia Retrieval (ICMR)* (Bucharest, Romania). Association for Computing Machinery, Bucharest, Romania, 278–286. https://doi.org/10.1145/3078971.3078979

[2] Luca Bondi, Silvia Lameri, David Güera, Paolo Bestagini, Edward J. Delp, and Stefano Tubaro. 2017. Tampering Detection and Localization Through Clustering of Camera-Based CNN Features. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. IEEE, Honolulu, HI, USA, 1855–1864. https://doi.org/10.1109/CVPRW.2017.232

[3] Antoni Buades, Bartomeu Coll, and Jean-Michel Morel. 2005. A non-local algorithm for image denoising. In *IEEE Conference on Computer Vision and Pattern Recognition*, Vol. 2. IEEE, San Diego, CA, USA, 60–65. https://doi.org/10.1109/CVPR.2005.38

[4] S. Grace Chang, Bin Yu, and Martin Vetterli. 2000. Adaptive wavelet thresholding for image denoising and compression. *IEEE Transactions on Image Processing* 9, 9 (2000), 1532–1546. https://doi.org/10.1109/83.862633

[5] Luca Cuccovillo and Patrick Aichroth. 2016. Open-set microphone classification via blind channel analysis. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, Shanghai, China, 2074–2078. https://doi.org/10.1109/ICASSP.2016.7472042

[6] Luca Cuccovillo, Sebastian Mann, Marco Tagliasacchi, and Patrick Aichroth. 2013. Audio tampering detection via microphone classification. In *IEEE International Workshop on Multimedia Signal Processing (MMSP)*. IEEE, Pula, Italy, 177–182.

[7] Alexandre Défossez, Gabriel Synnaeve, and Yossi Adi. 2020. Real Time Speech Enhancement in the Waveform Domain. In *Annual Conference of the International Speech Communication Association (Interspeech)*. ISCA, Shanghai, China, 3291–3295. https://doi.org/10.21437/Interspeech.2020-2409

[8] John S. Garofolo, Lori F. Lamel, William M. Fisher, Jonathan G. Fiscus, David S. Pallett, Nancy L. Dahlgren, and Victor Zue. 1993. TIMIT Acoustic-Phonetic Continuous Speech Corpus. https://doi.org/10.35111/17gk-bn40

[9] Nikolay D. Gaubitch, Mike Brookes, Patrick A. Naylor, and Dushyant Sharma. 2011. Single-microphone blind channel identification in speech using spectrum classification. In *IEEE European Signal Processing Conference (EUSIPCO)*. IEEE, Barcelona, Spain, 1748–1751.

[10] Antonio Giganti, Luca Cuccovillo, Paolo Bestagini, Patrick Aichroth, and Stefano Tubaro. 2022. Speaker-Independent Microphone Identification in Noisy Conditions. *arXiv preprint, submitted to EUSIPCO* (2022).

[11] Hynek Hermansky and Nelson Morgan. 1994. RASTA processing of speech. *IEEE Transactions on Speech and Audio Processing* 2, 4 (1994), 578–589. https://doi.org/10.1109/89.326616

[12] Constantine Kotropoulos and Stamatios Samaras. 2014. Mobile phone identification using recorded speech signals. In *IEEE International Conference on Digital*

SignalProcessing. IEEE, Hong Kong, China, 586–591. https://doi.org/10.1109/ICDSP.2014.6900732

[13] Christian Krätzer, Andrea Oermann, Jana Dittmann, and Andreas Lang. 2007. Digital Audio Forensics: A First Practical Evaluation on Microphone and Environment Classification. In *ACM Workshop on Multimedia & Security (MM&Sec)*. Association for Computing Machinery, Dallas, Texas, USA, 63–74. https://doi.org/10.1145/1288869.1288879

[14] Xiaodan Lin, Jianqing Zhu, and Donghua Chen. 2020. Subband Aware CNN for Cell-Phone Recognition. *IEEE Signal Processing Letters* 27 (2020), 605–609. https://doi.org/10.1109/LSP.2020.2985594

[15] Da Luo, Paweł Korus, and Jiwu Huang. 2018. Band Energy Difference for Source Attribution in Audio Forensics. *IEEE Transactions on Information Forensics and Security* 13, 9 (2018), 2179–2189. https://doi.org/10.1109/TIFS.2018.2812185

[16] Pedro Ribeiro Mendes Júnior, Luca Bondi, Paolo Bestagini, Stefano Tubaro, and Anderson Rocha. 2019. An In-Depth Study on Open-Set Camera Model Identification. *IEEE Access* 7 (2019), 180713–180726. https://doi.org/10.1109/ACCESS.2019.2921436

[17] Lyndon J.B. Nixon, Shu Zhu, Fabian Fischer, Walter Rafelsberger, Max Göbel, and Arno Scharl. 2017. Video Retrieval for Multimedia Verification of Breaking News on Social Networks. In *ACM First International Workshop on Multimedia Verification*. Association for Computing Machinery, Mountain View, California, USA, 13–21. https://doi.org/10.1145/3132384.3132386

[18] Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. 2015. Librispeech: An ASR corpus based on public domain audio books. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, South Brisbane, QLD, Australia, 5206–5210. https://doi.org/10.1109/ICASSP.2015.7178964

[19] Leonid I. Rudin, Stanley Osher, and Emad Fatemi. 1992. Nonlinear total variation based noise removal algorithms. *Physica D: Nonlinear Phenomena* 60, 1 (1992), 259–268. https://doi.org/10.1016/0167-2789(92)90242-F

[20] Carlo Tomasi and Roberto Manduchi. 1998. Bilateral filtering for gray and color images. In *IEEE International Conference on Computer Vision*. IEEE, Bombay, India, 839–846. https://doi.org/10.1109/ICCV.1998.710815

[21] Kai Zhang, Wangmeng Zuo, Yunjin Chen, Deyu Meng, and Lei Zhang. 2017. Beyond a Gaussian denoiser: Residual learning of deep CNN for image denoising. *IEEE Transactions on Image Processing* 26, 7 (2017), 3142–3155.