Multimedia Forensics: From Statistical Signal Processing to Deep Learning and Explainable Artificial Intelligence

Constantine Kotropoulos

Department of Informatics, Aristotle University of Thessaloniki, GREECE

2025 Int. Conf. on Advances in Databases, Knowledge, and Data Applications (InfoSys)

March 9, 2025

ARISTOTLE UNIVERSITY OF THESSALONIKI

Acknowledgments

Christos Korgialas, Ioannis Tsingalis, Georgios Tzolopoulos, Dimitrios Kritsiolis, Prof. Yannis Panagakis, Stamatios Samaras, Ruizhe Li, Prof. Chang-Tsun Li, and Prof. Yu Guan

Presenter's bio

Konstantinos (Constantine) Kotropoulos Konstantinos received the Diploma degree with honors in Electrical Engineering in 1988 and the PhD degree in Electrical and Computer Engineering in 1993, both from the Aristotle University of Thessaloniki. He is currently a Full Professor in the Department of Informatics at the Aristotle University of Thessaloniki. He was a visiting research scholar in the Department of Electrical and Computer Engineering at the University of Delaware, USA during the academic year 2008-2009 and he conducted research in the Signal Processing Laboratory at Tampere University of Technology, Finland during the summer of 1993. He has co-authored 71 journal papers, 226 conference papers, and contributed 9 chapters to edited books in his areas of expertise. He is co-editor with Professor Ioannis Pitas of the book "Nonlinear Model-Based Image/Video Processing and Analysis" (J. Wiley and Sons, 2001).

Dr. Kotropoulos has made fundamental contributions to speech emotion recognition. His research connects several science disciplines, such as efficient feature selection and multivariate statistics. He has also contributed to non-linear signal processing, including non-linear filters resorting to linear combinations of order statistics, music genre recognition exploiting spatio-temporal modulations, non-negative tensor decompositions, as well as sparse and low-rank representations, or using alpha-stable projections to derive sketches of features that captured the intrinsic trace of the acquisition device.

He is a fellow of IARIA, a senior member of the IEEE, and a member of EURASIP and IAPR. He was a Senior Area Editor of the IEEE Signal Processing Letters and he has been a member of the Editorial Board of another 7 journals. He served as Track Chair for Signal Processing in the 6th Int. Symposium on Communications, Control, and Signal Processing, Athens, 2014; Program Co-Chair of the 4th Int. Workshop on Biometrics and Forensics, Limassol, Cyprus, 2016; Technical Program Chair of the XXV European Signal Processing Conf., Kos, Greece, 2017; Technical Program Chair of the 5th IEEE Global Conf. Signal and Information Processing, Montreal, Canada, 2017; General Chair of the 2022 IEEE 14th Image, Video and Multidimensional Signal Processing Workshop, Nafplio, Greece; Technical Program Chair of the 2023 IEEE Int. Conf. on Acoustics, Speech, and Signal Processing, Rhodes, Greece.

Outline

Threat of knowledge life cycle

- Digital speech content can be imperceptibly altered by malicious, even amateur, users employing a variety of low-cost audio editing software.
- This threat permeates a wide variety of fields, such as intellectual property, intelligence gathering, forensics, news reporting, etc.
- Theories and tools to combat this threat in the digital speech forensics are still in their infancy.
- A first step to combat this threat is to extract forensic evidence about the mechanism involved in the generation of the speech recording. That is, to *identify the acquisition device*.
- Other problems in digital speech forensics are: codec identification, authentication of speakers' environment, identification of the device power source (i.e., electric network frequency), identification of the network traversed.

Intrinsic traces

- Suppose that the device along with its associated signal processing chain leaves behind intrinsic traces in the speech signal.
- Various devices (e.g., telephone handsets, mobile phones) do not have exactly the same frequency response due to the tolerance in the nominal values of the electronic components and the different designs employed by the various manufacturers.
- A blind-passive approach is exploited, as opposed to active embedding of watermarks or having access to input-output pairs.

Mobile Phone Identification

Intrinsic traces based on MFCCs

- Mel frequency cepstral coefficients (MFCCs) are extracted from any recorded speech signal at a frame level.

Mobile Phone Identification

Intrinsic traces based on MFCCs

- Mel frequency cepstral coefficients (MFCCs) are extracted from any recorded speech signal at a frame level.
- The MFCC calculation employs frames of duration 20 ms with a hop size of 10 ms, and a 42-band filter bank. The correlation between the frequency bands is reduced by applying the discrete cosine transform to the log energies of the bands.

Mobile Phone Identification

Intrinsic traces based on MFCCs

- Mel frequency cepstral coefficients (MFCCs) are extracted from any recorded speech signal at a frame level.
- The MFCC calculation employs frames of duration 20 ms with a hop size of 10 ms, and a 42-band filter bank. The correlation between the frequency bands is reduced by applying the discrete cosine transform to the log energies of the bands.
- Gaussian Mixture Models (GMMs) with diagonal covariance matrices are used in order to model the probability density function of the MFCC vectors.

Mobile Phone Identification

Intrinsic traces based on MFCCs

- Mel frequency cepstral coefficients (MFCCs) are extracted from any recorded speech signal at a frame level.
- The MFCC calculation employs frames of duration 20 ms with a hop size of 10 ms, and a 42-band filter bank. The correlation between the frequency bands is reduced by applying the discrete cosine transform to the log energies of the bands.
- Gaussian Mixture Models (GMMs) with diagonal covariance matrices are used in order to model the probability density function of the MFCC vectors.
- A Gaussian supervector (GSV) derived by concatenating the mean vectors and the main diagonals of the covariance matrices is used as a template for each device.

Mobile Phone Identification

MOBIPHONE database

- The MOBIPHONE database contains 21 mobile phones of various models from 7 different brands.
- 12 male and another 12 female speakers were randomly chosen from TIMIT[1].
- Each speaker reads 10 sentences approximately 3s long.
- The first 2 sentences are the same for every speaker, but the rest are different.

[1] J. Garofolo, "Getting started with the DARPA TIMIT CD-ROM: An acoustic phonetic continuous speech database," National Inst. Standards and Technology (NIST), Tech. Rep., 1988.

Mobile Phone Identification

MOBIPHONE database

- Sampling frequency: 16 kHz.
- The recordings were made in a silent, controlled environment with the same recording equipment.
- The 10 utterances per speaker were concatenated in a single 30s long recording, yielding 504 recordings all together.
- MOBIPHONE is publicly available[2].
- A second version of MOBIPHONE, where each recording is split into its constituent utterances is also available[3].

[2] https://www.dropbox.com/sh/9n7fy7moi825bgk/WFLBKxUitV

[3] https://tinyurl.com/4kmb6fj7.

Table 1: Brands and models of the mobile phones in the MOBIPHONE and their class names.

Class Name	Brand and Model	Class Name	Brand and Model
HTC1	HTC desire c	APPLE1	iPhone5
HTC2	HTC sensation xe	S1	Samsung E2121B
LG1	LG GS290	S2	Samsung E2600
LG2	LG L3	S3	Samsung GT-I8190 mini
LG3	LG Optimus L5	S4	Samsung GT-N7100 (Note2)
LG4	LG Optimus L9	S5	Samsung Galaxy GT-I9100 s2
N1	Nokia 5530	S6	Samsung Galaxy Nexus S
N2	Nokia C5	S7	Samsung e1230
N3	Nokia N70	S8	Samsung s5830i
SE1	Sony Ericsson c902	V1	Vodafone joy 845
SE2	Sony Ericsson c510i		

Mobile Phone Identification

Power spectrum

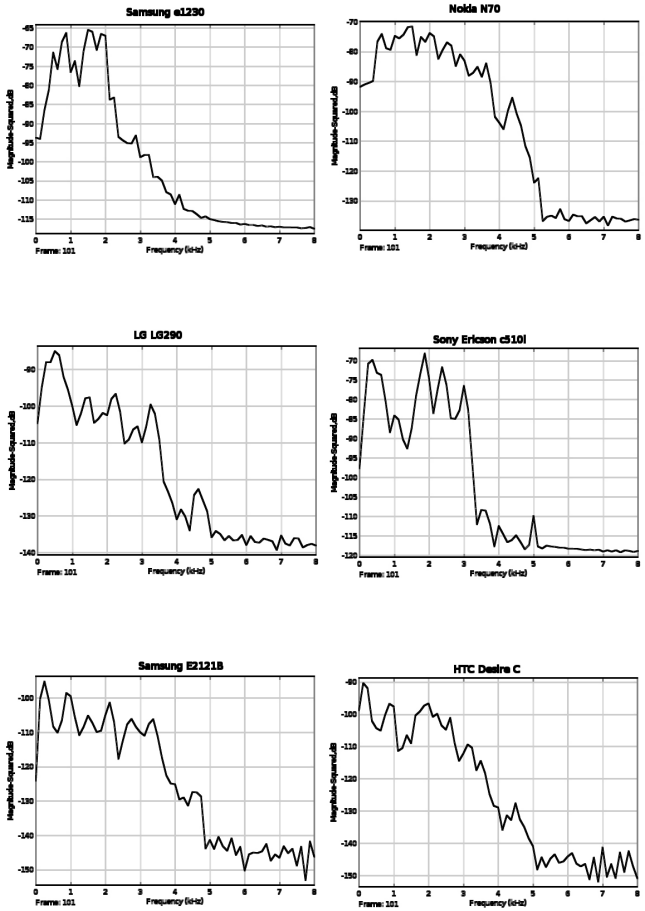


Figure 1: Power spectrum of the same speech signal recorded by different mobile phones.

Mobile Phone Identification

Histograms of the 5th MFCC

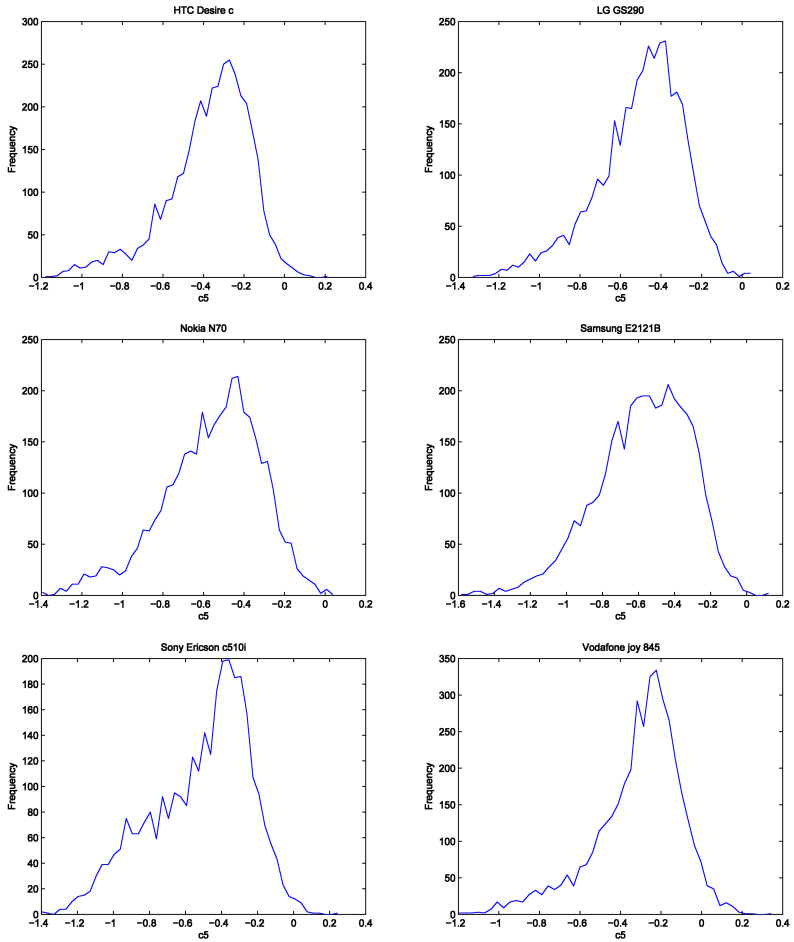


Figure 2: Histogram of the 5th MFCC when the same speech signal is recorded by different mobile phones.

Mobile Phone Identification

Experimental Results

- The first set of experiments employs GSVs without GMM-Universal Background Model (UBM) and three commonly used classifiers, such as the Support Vector Machines (SVMs) with different kernels, a Radial Basis Function (RBF)-NN, and a Multilayer Perceptron (MLP)[4].
- The second set of experiments employs sketches of GSVs with GMM-UBM and three classifiers, namely, the SRC, the linear SVMs, and the Nearest Neighbor (NN) classifier[5].

[4] C. Kotropoulos and S. Samaras, "Mobile phone brand and model identification using recorded speech signals," in *Proc. 19th Int. Conf. Digital Signal Processing*, pp. 586-591, August 20-23, 2014.

[5] C. Kotropoulos, "Source phone identification using sketches of features," *IET Biometrics*, vol. 4, no. 2, pp. 75-83, June 2014.

Mobile Phone Identification

Experimental Results

Table 2: Identification accuracies (in %) achieved by the RBF-NN for GSVs including also the variances.

Components	σ	Accuracy
1	0.1	97.6
3	5	73.8
6	5	74.2

Mobile Phone Identification

Experimental Results

Table 3: Identification accuracies (in %) achieved by the MLP for both types of GSVs (M stands for the GSVs formed by the mean vectors only and MC stands for the GSVs, which include the variances as well).

Components	GSV Type	Momentum	Learning Rate	Hidden Neurons	Epochs	Accuracy
1	M	0.9	0.1	150	250	96
2	MC	0.9	0.1	150	250	96.4

Telephone Handset Identification

Lincoln-Labs handset database (LLHDB)

- A subset of LHHDB[6] was used that consisted of speech recordings from 53 speakers (24 males and 29 females) acquired by 8 landline telephone handsets.
- 4 of telephone handsets are carbon-button (CB1-CB4) and 4 are electrect (EL1-EL4).
- Evaluation Protocol: stratified 2-fold cross-validation.
- Baseline Classifiers: Linear SVM, NN with cosine similarity measure.

[6] D. A. Reynolds, "HTIMIT and LLHDB: speech corpora for the study of handset transducer effects," in *Proc. 1997 IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, Munich, Germany, 1997, vol. 2, pp. 1535–1538.

Telephone Handset Identification

Average log spectrograms

- The spectrogram of each recorded speech signal is calculated by employing frames of duration 64 ms with a hop size of 32 ms.
- The logarithm of the spectrogram is calculated and averaged along the time axis, yielding a 2048-dimensional average log spectrogram.

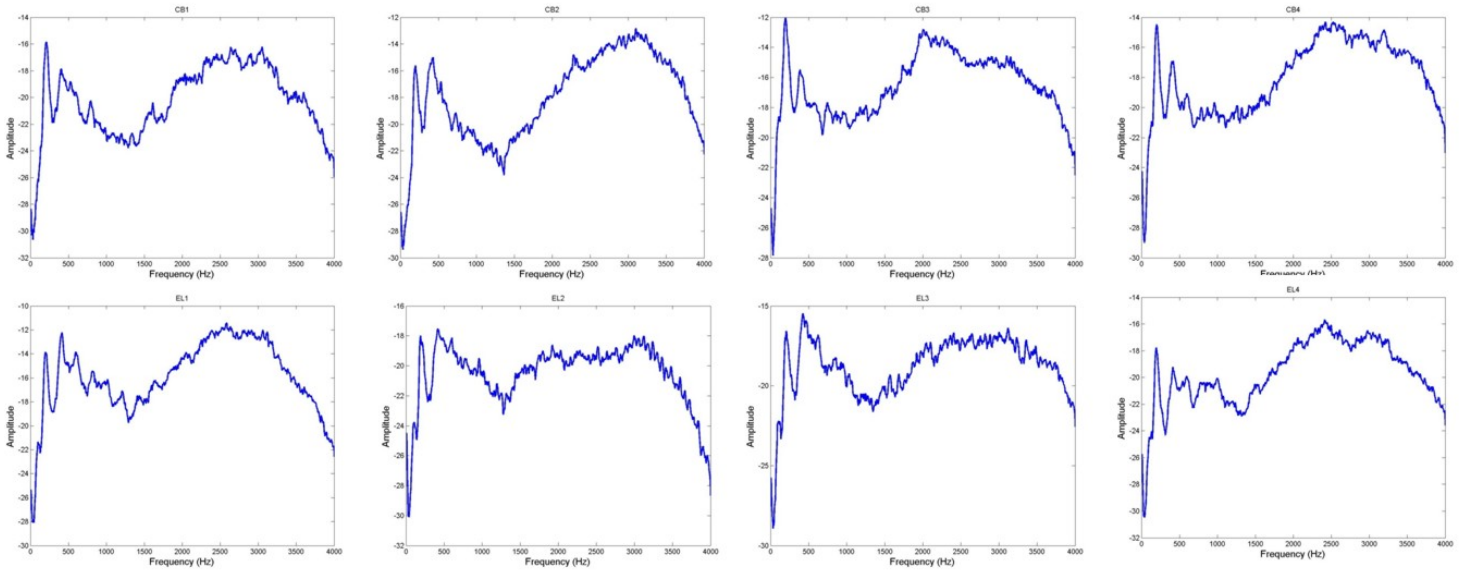


Figure 3: Average log spectrogram of a speech utterance recorded by 8 different telephone handsets in LLHDB.

Telephone Handset Identification

Sketches of spectral features (SSFs)

- The spectrogram of each recorded speech signal is calculated by employing frames of duration 64 ms with a hop size of 32 ms.
- The logarithm of the spectrogram is calculated and averaged along the time axis, yielding a 2048-dimensional average spectrogram.
- Let $\boldsymbol{Z} \in \mathbb{R}^{2048 \times n}$ be the data matrix containing the average log spectrograms of n recordings. The dimensionality of the average log spectrogram is reduced to $d < 2048$ by premultiplying $\boldsymbol{Z}$ with a projection matrix $\boldsymbol{R} \in \mathbb{R}^{d \times 2048}$ yielding $\boldsymbol{X} = \boldsymbol{R}\,\boldsymbol{Z}$.
- The elements of $\boldsymbol{R}$, $R_{i,j}$ are taken as independent identically distributed (i.i.d.) random variables sampled from a p-stable distribution.
- The resulting d-dimensional SSFs are used for acquisition device representation.

Telephone Handset Identification

Sketches of spectral features (SSFs)

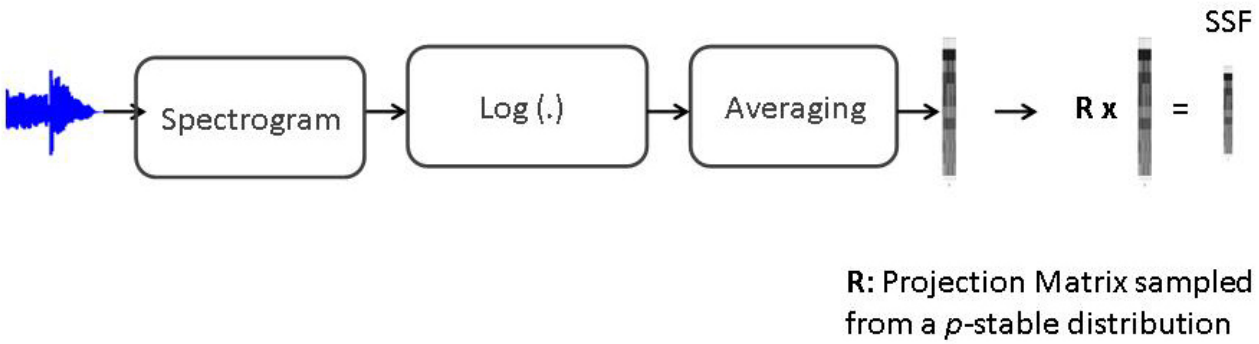


Figure 4: Unsupervised feature selection: **Sketches of Spectral Features**.

Telephone Handset Identification

p-stable distributions

- If we sample $R_{i,j}$ from a p-stable distribution, for any two mean spectrograms (say the first two column of Z) the differences $X_{i,1} - X_{i,2} = \sum_{j=1}^{2048} R_{i,j} (Z_{j,1} - Z_{j,2})$, $i = 1, 2, \ldots, d$, are also i.i.d. samples of a p-stable distribution.
- The projection can be used to recover an approximate value of the ℓ_p norm of the original spectrograms computed in a space of reduced dimensions.
- The most well-known stable distribution is the Gaussian distribution of zero mean and unit standard deviation, which is 2-stable. It was used to extract the Random Spectral Features (RSFs).
- The class of stable distributions is much wider, including heavy-tailed distributions as well[7].

- For $p = 0.5$, one obtains the Levy distribution. The Cauchy distribution, $f(r) = \frac{1}{\pi}\frac{1}{1+r^2}$, is 1-stable. The aforementioned three distributions are the only cases for which closed form expressions of the probability density functions exist[8].

[7] P. Indyk, "Stable distributions, pseudorandom generators, embeddings, and data stream computation," *Journal of the ACM*, vol. 53, no. 3, pp. 307–323, 2006.

[8] G. R. Arce, *Nonlinear Signal Processing*, J. Wiley & Sons, Hoboken, NJ, USA, 2005.

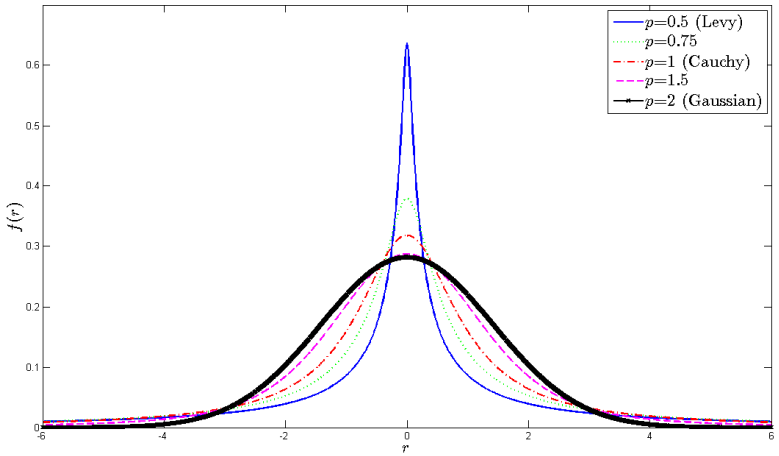


Figure 5: *p*-stable probability density functions.

- In general for $p \in (0, 2]$, $R_{i,j}$ can be generated by[9]

$$R_{i,j} = \frac{\sin(p\theta)}{\cos^{1/p}\theta} \left(\frac{\cos(\theta(1-p))}{-\ln u} \right)^{\frac{1-p}{p}}$$

where θ is uniform on $[-\pi/2, \pi/2]$ and u is uniform on $[0, 1]$.

[9] J. P. Nolan, *Stable Distributions*, Birkhauser, 2002.

Telephone Handset Identification

Labeled spectral features (LSF)

- Let $\boldsymbol{X}_t \in \mathbb{R}^{2048 \times N_t}$ be the training data matrix, containing in its columns the average log spectrograms extracted from N_t speech signals recorded by using acquisition devices from K classes.
- Let $\boldsymbol{L}_t \in \{0,1\}^{K \times N_t}$ be the label indicator matrix, where the kth component of the nth column of $\boldsymbol{L}_t$, $\boldsymbol{l}_n$, is 1 if the nth device belongs to class $k \in \mathcal{K} = \{1, 2, \ldots, K\}$.
- Features highly dependent on the labels can be obtained by seeking a linear mapping $\boldsymbol{M} \in \mathbb{R}^{K \times 2048}$ such that the space of the training average log spectrograms is mapped onto the label space, i.e., $\boldsymbol{L}_t = \boldsymbol{M}\,\boldsymbol{X}_t$.

- $\boldsymbol{M}$ is found by solving the following ridge regression problem:

$$\underset{\boldsymbol{M}}{\operatorname{argmin}} \|\boldsymbol{L}_t - \boldsymbol{M}\,\boldsymbol{X}_t\|_F^2 + \lambda\|\boldsymbol{M}\|_F^2, \tag{1}$$

  where λ is a regularization parameter ($\lambda = 0.5$ was used in the experiments) and $\|.\|_F$ denotes the Frobenius norm.
- The unique closed form solution of (1) is

$$\boldsymbol{M} = \boldsymbol{L}_t\,\boldsymbol{X}_t^\top \left(\boldsymbol{X}_t\,\boldsymbol{X}_t^\top + \lambda\,\boldsymbol{I}\right)^{-1}. \tag{2}$$

- In the test phase, by premultiplying any average log spectrogram by $\boldsymbol{M}$, the K-dimensional vector of the LSFs is obtained.

Telephone Handset Identification

Sparse-representation based classifier

- Let $\boldsymbol{X} = [\boldsymbol{X}_1|\boldsymbol{X}_2|\ldots|\boldsymbol{X}_K] \in \mathbb{R}^{d\times n}$ be an overcomplete dictionary formed by concatenating n intrinsic traces (e.g., average MFCCs, RSFs, SSFs, LSFs), which stem from K acquisition devices[10].
- We assume a vector of test intrinsic traces $\boldsymbol{y} \in \mathbb{R}^d$ comes from the ith device. Let $\boldsymbol{c} = \left[\boldsymbol{0}^\top|\ldots|\boldsymbol{0}^\top|\ \boldsymbol{c}_i^\top|\boldsymbol{0}^\top|\ldots|\boldsymbol{0}^\top\right]^\top$ be the $n \times 1$ augmented coefficient vector, whose elements are zero except those associated with the ith device. $\boldsymbol{y}$ is expressed as a linear combination of the atoms that are associated to the ith device, i.e., $\boldsymbol{y} = \boldsymbol{X}\,\boldsymbol{c}$.

- Since the device ID of test vector $\boldsymbol{y}$ is unknown, we predict it by seeking the sparsest solution to the linear system of equations $\boldsymbol{y} = \boldsymbol{X}\,\boldsymbol{c}$. i.e.,

$$\boldsymbol{c}^* = \arg\min_{\boldsymbol{c}} \|\boldsymbol{c}\|_1 \quad \text{subject to } \boldsymbol{X}\,\boldsymbol{c} = \boldsymbol{y}, \tag{3}$$

  where $\|.\|_1$ denotes the ℓ_1 norm of a vector.
- $\boldsymbol{y}$ is classified to the device class that minimizes the residual $r_i(\boldsymbol{y}) = \|\boldsymbol{y} - \boldsymbol{X}\,\delta_i(\boldsymbol{c})\|_2$, where $\delta_i(\boldsymbol{c}) \in \mathbb{R}^n$ is a vector, whose nonzero entries are associated to the ith device only.

Telephone Handset Identification

Sparse-representation based classifier

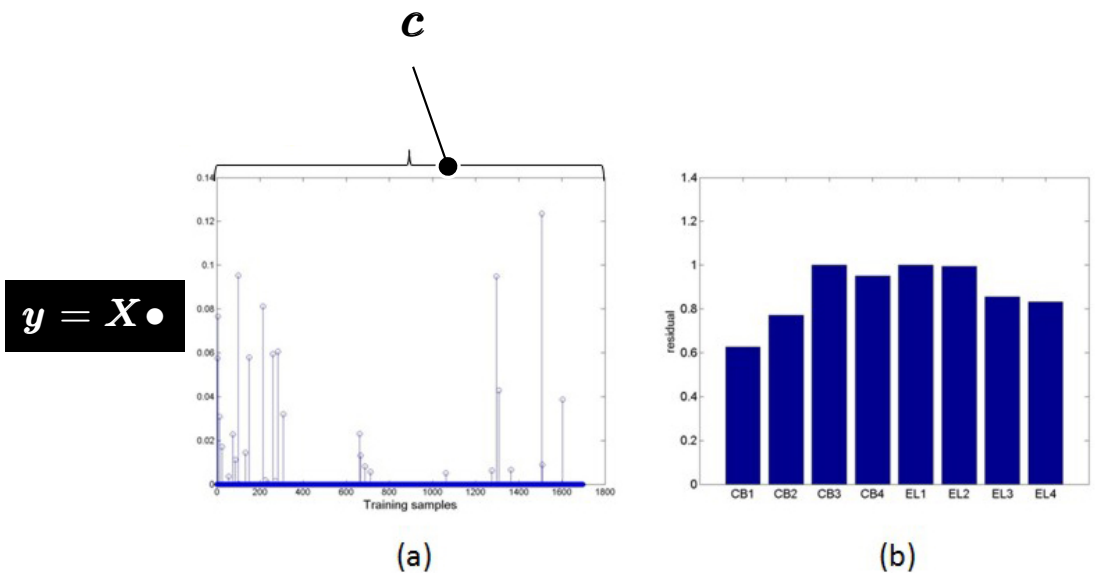


Figure 6: The test vector of RSFs $\boldsymbol{y}$ has been extracted by a carbon-button telephone handset with the ID: CB1. (a) Sparse coefficients $\boldsymbol{c}$. The non-zero entries of $\boldsymbol{c}$ are mainly associated with RSFs extracted from speech utterances recorded with the CB1. (b) The residuals $r_i(\boldsymbol{y})$ of the RSFs. The smallest residual value reveals the identity of the telephone handset (i.e., CB1).

[10] J. Wright, A. Yang, A. Ganesh, S. Sastry, and Y. Ma, "Robust face recognition via sparse representation," *IEEE Trans.Pattern Analysis and Machine Intelligence*, vol. 31, no. 2, pp. 210–227, 2009.

Telephone Handset Identification

Experiments on LHHDB

Table 4: Best telephone handset identification accuracies achieved by the SSFs and the MFCCs, when the SRC, the linear SVM, and the NN are employed.

Features	**Feature dimension**	**Classifier**	**Accuracy (%)**
SSFs (Cauchy)	800	SRC	**94.72**
SSFs (Cauchy)	800	SVM	94.66
SSFs (Cauchy)	775	NN	83.78
SSFs (Gaussian)	700	SRC	**94.99**
SSFs (Gaussian)	800	SVM	94.66
SSFs (Gaussian)	850	NN	85.08
LSFs	8	SRC	**97.14**
LSFs	8	SVM	**97.58**
LSFs	8	NN	96.52
MFCCs	23	SRC	89.79
MFCCs	23	SVM	87.35
MFCCs	23	NN	81.95
MFCC-based Gaussian supervector [11]	N/A	SVM	93.20

[11] D. Garcia-Romero and C. Y. Espy-Wilson, "Automatic acquisition device identification from speech recordings," in *Proc. 2010 IEEE Int. Conf. Acoustics, Speech, and Signal Processing, Dallas, TX, USA, pp. 1806-1809, 2010.*

Revisiting Mobile Phone Identification

Second Set of Experiments on MOBIPHONE

Table 5: Best mobile phone identification accuracies achieved by the SRC, the linear SVM, and the NN for various sketches of features.

Features	**Classifier**	**Accuracy (%)**
GSVs using a GMM-UBM trained on the full training subset of TIMIT for 64 or 128 GMM components	SRC	100
	SVM	100
	NN	100
GSVs using a GMM-UBM trained on the training subset DR2 of TIMIT for 64 GMM components	SRC	100
	SVM	100
	NN	99.60
GSVs using a GMM-UBM trained on the training subset DR2 of TIMIT for 128 GMM components	SRC	100
	SVM	100
	NN	98.41

Revisiting Mobile Phone Identification

Second Set of Experiments on MOBIPHONE

Table 5: Best mobile phone identification accuracies achieved by the SRC, the linear SVM, and the NN for various sketches of features (cont.).

Features	**Classifier**	**Accuracy (%)**
GSV with 1 GMM component trained on the MOBIPHONE	SRC	99.21
	SVM	98.41
	NN	98.02
GSV with 2 GMM components trained on the MOBIPHONE	SRC	96.83
	SVM	95.24
	NN	94.05
SSFs	SRC	98.81
	SVM	96.03
	NN	94.84
average MFFCs	SRC	96.82
	SVM	97.22
	NN	96.03

- We examine two tasks: brand identification and model identification.
- A 3-stage process describes mobile phone identification:
  1. Feature extraction: MFCCs.
  2. Feature modeling: GMMs and GSVs.
  3. Classification: Maximum Likelihood (ML), SVMs, Neural models.

- ML classification using the GMM probability density function as a similarity measure between the test utterance and the brand or model the GMM represents.
- SVM classification using the GSVs of the utterances.
- Neural classification with an architecture specifically designed for mobile phone identification[12] that uses both the MFCCs and the GSVs as input.

[12] C. Zeng, S. Feng, D. Zhu, and Z. Wang, "Source acquisition device identification from recorded audio based on spatiotemporal representation learning with multi-attention mechanisms," *Entropy*, vol. 25(4):626.

Alternative approaches

- The process discussed thus far will be referred to as the traditional method.
- We examine two alternative approaches to the traditional method.
  1. Non-speech approach: Extracts the MFCCs from the non-speech parts of the signals.
  2. UBM approach: Obtains the GMMs from the Maximum A Posteriori (MAP) adaptation of a UBM. A UBM is a large speaker-independent GMM trained via the EM algorithm to capture the distribution of the MFCCs in an extensive set of speakers.

Non-Speech Segments and UBM Adaptation

Non-speech approach

- Speech parts have speaker-dependent information irrelevant to the task at hand.
- The noise-like signals in the non-speech parts have a flatten spectral density and better capture the mobile phone's recording circuitry transfer function.
- Non-speech MFCCs yield higher mutual information than the MFCCs obtained from the entire speech recording[13].
- Voice activity detection[14] is used to obtain non-speech frames.

[13] C. Hanilci and T. Kinnunen, "Source cell-phone recognition from recorded speech using non-speech segments," *Digital Signal Processing*, vol. 35, pp. 75–85, 2014

[14] J. Sohn, N. S. Kim, "A statistical model-based voice activity detection," *IEEE Signal Processing Letters*, vol. 6, no. 1, pp. 1–3., 1999.

- Thus, speaker recognition techniques, such as a UBM[15], are justified.
- Utterance-specific GMMs are obtained from the UBM via MAP adaptation of the utterance's MFCC vectors.
- GSVs are obtained by concatenating only the mean vectors of the UBM-adapted GMMs.

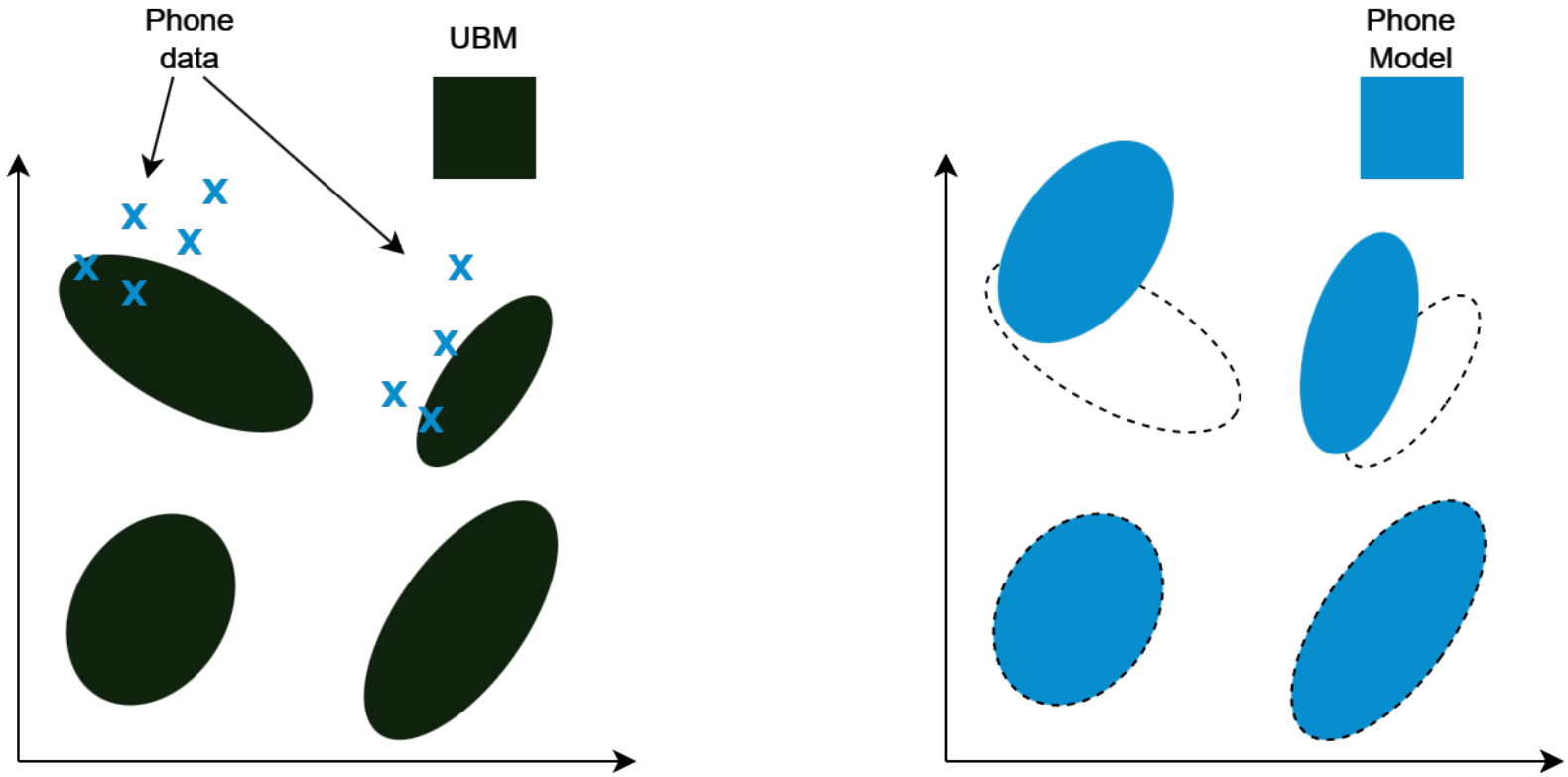


Figure 7: MAP adaptation of a UBM.

[15] T. Reynolds, T. Quatieri, and R. Dunn, "Speaker verification using adapted Gaussian mixture models," *Digital Signal Processing*, vol. 10, no:1-3, pp. 19–41, 2000.

MOBIPHONE dataset

- It consists of 10 recorded speech utterances of 24 speakers, 12 males, and 12 females, using 21 mobile phones belonging to 7 brands. The 24 speakers were randomly chosen from the TIMIT dataset.
- Two sets of augmentations were applied to the dataset to compare the relative performance of the three methods under noisy recording conditions and amplified speaker variability:
  1. Recording conditions: Gaussian noise, background noise, and reverberation.
  2. Speaker variability: Croppings, loudness, pitch, speed, and Vocal Tract Length Perturbation (VTLP).

The Central China Normal University (CCNU) mobile dataset

- The CCNU Mobile dataset[16] consists of 642 speech recordings from the TIMIT dataset recorded by 45 mobile phones belonging to 9 brands. Multiple devices of the same mobile phone model are included in the dataset.
- Unfortunately, the entire dataset is not publicly available. We were able to use a small subset of the dataset, which consists of 5 8-second-long audio samples of female speakers per mobile phone.
- This data scarcity will allow us to test the different methods under realistic audio forensic conditions with limited data.

[16] C. Zeng, D. Zhu, Z. Wang, Z. Wang, N. Zhao, and L. He, "An end-to-end deep source recording device identification system for web media forensics," *International Journal of Web Information Systems*, vol. 16, no. 4, pp. 413–425, 2020.

Non-Speech Segments and UBM Adaptation

MOBIPHONE Results

Table 6: ML classification accuracy for brand and model identification on the MOBIPHONE dataset using audio and non-speech MFCCs.

	Brand		Model	
# Components	Audio	Non-speech	Audio	Non-speech
1	66.6667%	79.3651%	96.0317%	98.8095%
2	73.4127%	86.1111%	98.8095%	99.6032%
4	88.4921%	96.8254%	98.4127%	100%
8	91.6667%	99.2063%	99.2063%	100%
16	88.0952%	99.6032%	98.4127%	100%
32	93.254%	100%	98.8095%	100%
64	98.0159%	99.6032%	98.4127%	100%

Non-Speech Segments and UBM Adaptation

MOBIPHONE Results

Table 7: The best classification accuracy on MOBIPHONE for model identification on the first set of augmentations.

	Audio		Non-speech		UBM
	ML	SVM	ML	SVM	SVM
Baseline	$99.2063\%^{8}$	$96.4286\%^{1/MC}$	$\mathbf{100\%}^{4}$	$99.2063\%^{1/M}$	$98.8095\%^{4/M}$
Gaussian	$89.881\%^{16}$	$90.6746\%^{1/MC}$	$78.9683\%^{64}$	$\mathbf{93.8492\%}^{1/M}$	$92.8571\%^{2/M}$
Background	$93.0556\%^{16}$	$95.4365\%^{1/MC}$	$83.3333\%^{64}$	$94.0476\%^{1/MC}$	$\mathbf{97.4206\%}^{2/M}$
Reverberation	$98.6111\%^{8}$	$96.2302\%^{2/MC}$	$\mathbf{99.6032\%}^{32}$	$98.4127\%^{1/M}$	$98.2143\%^{4/M}$
All	$91.5675\%^{64}$	$92.2619\%^{1/M}$	$80.2579\%^{64}$	$91.8651\%^{1/MC}$	$\mathbf{96.2302\%}^{2/M}$

Non-Speech Segments and Universal Background Model Adaptation

MOBIPHONE Results

Table 8: The best classification accuracy on MOBIPHONE for model identification on the second set of augmentations.

	Audio		Non-speech		UBM
	ML	SVM	ML	SVM	SVM
Baseline	$99.2063\%^{8}$	$96.4286\%^{1/MC}$	$\mathbf{100\%}^{4}$	$99.2063\%^{1/M}$	$98.8095\%^{4/M}$
Crop	$99.4048\%^{32}$	$96.2302\%^{2/MC}$	$\mathbf{100\%}^{4}$	$99.0079\%^{1/MC}$	$99.6032\%^{4/M}$
Loudness	$98.8095\%^{8}$	$97.4206\%^{1/MC}$	$\mathbf{99.2063\%}^{8}$	$99.0079\%^{1/M}$	$99.0079\%^{2/M}$
Pitch	$94.6429\%^{64}$	$86.9048\%^{1/MC}$	$\mathbf{98.4127\%}^{32}$	$97.0238\%^{1/MC}$	$97.4206\%^{4/M}$
Speed	$98.8095\%^{8}$	$96.627\%^{1/MC}$	$\mathbf{100\%}^{4}$	$99.4048\%^{1/MC}$	$98.0159\%^{8/M}$
VTLP	$\mathbf{99.4048\%}^{64}$	$98.0159\%^{1/MC}$	$98.8095\%^{4}$	$99.2063\%^{1/MC}$	$98.8095\%^{4/M}$
All	$97.2884\%^{64}$	$93.1217\%^{1/M}$	$\mathbf{98.8095\%}^{32}$	$96.6931\%^{1/M}$	$\mathbf{98.8095\%}^{8/M}$

Non-Speech Segments and Universal Background Model Adaptation

CCNU Mobile Results

Table 9: Mean and standard deviation of the classification accuracy for model identification using the three approaches with ML, SVMs, and the neural network proposed by (Zeng et al., 2023).

	Audio	Non-speech	UBM
ML-64	98.5185% (0%)	98.5185% (0%)	-
SVM-1M	90.8148% (5.7292%)	96.5185% (3.5447%)	-
SVM-1MC	81.9259% (6.3356%)	90.7407 % (1.0030%)	-
SVM-4M	-	-	93.3333% (0.7808%)
MFCC branch	22.3703% (5.2471%)	43.7037% (7.7454%)	21.7037% (5.3307%)
GSV branch	56.7407% (4.3361%)	89.8518% (2.9845%)	98.2222% (1.1152%)
Fusion branch	52.1481% (2.0717%)	86.4444% (4.3199%)	63.5555% (13.1018%)

Source Camera Identification

Sensor Pattern Noise

- Sensor Pattern Noise (SPN), produced by imaging sensors, is a fingerprint of imaging devices linking photographs to the cameras acquired them. SPN can identify both the brand and the model of the camera. It is primarily caused by the different sensitivity of individual sensor pixels to light. SPN captures the intensity variations of individual pixels. It is a unique pattern deposited in every image.
- SPN is usually has a very high dimension (e.g., 1024×1024 pixels). However, such a high-dimensional feature is more likely to introduce redundancy and interfering components. For example, the extracted SPN can be contaminated by colour interpolation, JPEG compression, distortion introduced by denoising filter, and other artifacts. Most of these artifacts are non-unique, redundant, and less discriminant.
- To reduce SPN dimension, Principal Component Analysis (PCA) denoising was applied [17].

- However, the PCA denoising method is vulnerable when the training samples are seriously affected by the image content. Accordingly, it is difficult to train a reliable feature extractor by using a polluted training set.
- A camera identification framework based on the Random Subspace Method (RSM) and Majority Voting (MV) was proposed to solve this problem[18].

[17] R. Li, C. -T. Li, and Y. Guan, "A compact representation of sensor fingerprint for camera identification and fingerprint matching," in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, Brisbane, Australia, Apr. 19-24, 2015, pp. 1777–1781.

[18] R. Li, C. Kotropoulos, C. -T. Li, and Y. Guan, "Random Subspace Method For Source Camera Identification," in *Proc. IEEE Int. Workshop Machine Learning for Signal Processing*, Boston, MA, September 17-20, 2015.

Source Camera Identification

Sensor Pattern Noise

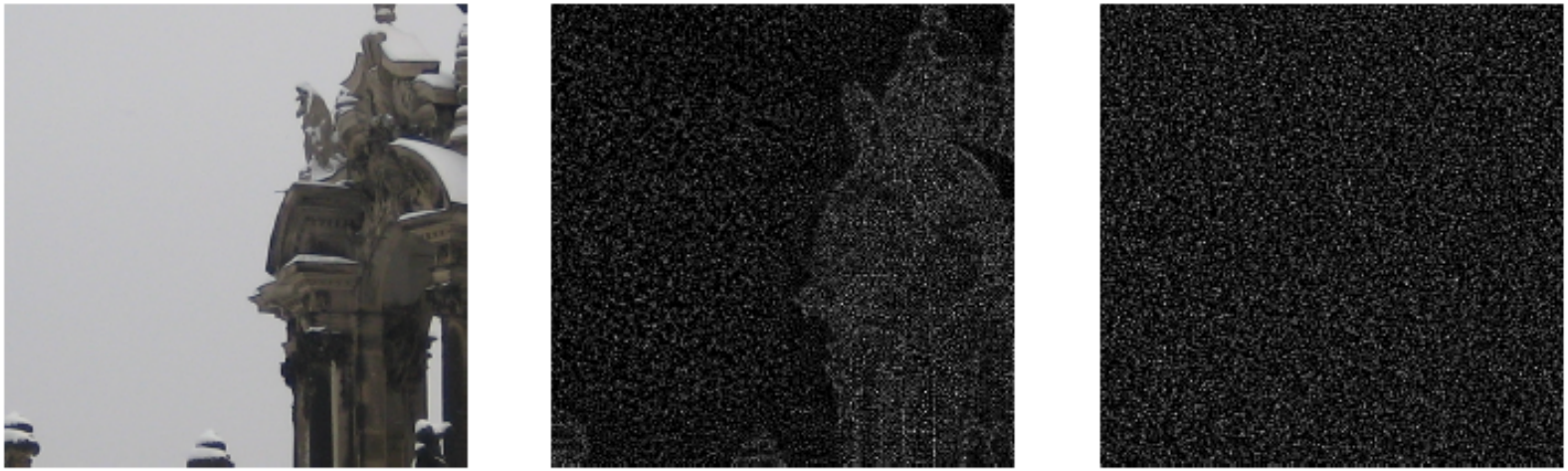


Figure 8: (a) An image taken by Canon Ixus70; (b) the noise residual extracted from (a), which is contaminated by scene details; (c) clean reference SPN of Canon Ixus70. (Note the intensity has been scaled to the interval [0,255] for visualization purposes.)

- Assume there are n images $\{I_i\}_{i=1}^{n}$, taken by c cameras $\{C_j\}_{j=1}^{c}$. Each camera has captured E_j images. First the noise residual is extracted from the $N \times N$-pixel blocks cropped from the center of the full-sized images, yielding $\{\boldsymbol{x}_i \in \mathbb{R}^{N \times N}\}_{i=1}^{n}$ which are centered. These n SPN templates are used as the training set. PCA is performed to seek a set of orthonormal vectors $\boldsymbol{v}_k$ and their associated eigenvalues λ_k of the autocorrelation matrix $\boldsymbol{S} \in \mathbb{R}^{N^2 \times N^2}$

$$\boldsymbol{S} = \frac{1}{n} \sum_{i=1}^{n} \boldsymbol{x}_i \, \boldsymbol{x}_i^{\top} = \boldsymbol{A} \boldsymbol{A}^{\top} \tag{4}$$

where $\boldsymbol{A} = \frac{1}{\sqrt{n}} [\boldsymbol{x}_1 | \boldsymbol{x}_2 | \ldots | \boldsymbol{x}_n] \in \mathbb{R}^{N^2 \times n}$.

- To make PCA feasible for the high-dimensional SPN, the $n \ll N^2$ eigenvectors of $\boldsymbol{A}^\top \boldsymbol{A} \in \mathbb{R}^{n \times n}$ are computed. Assume $\boldsymbol{v}'_k$ is the unit eigenvector of $\boldsymbol{A}^\top \boldsymbol{A}$ with eigenvalue λ'_k. The eigenvector $\boldsymbol{v}_k$ of $\boldsymbol{S}$ is obtained by $\boldsymbol{v}_k = \boldsymbol{A}\boldsymbol{v}'_k$ which is associated to eigenvalue $\lambda_k = \lambda'_k$.
- The feature space is derived $\boldsymbol{T} = [\boldsymbol{v}_1 | \boldsymbol{v}_2 | \ldots | \boldsymbol{v}_q] \in \mathbb{R}^{N^2 \times q}$ by retaining the eigenvectors associated to the top $q = \min_{q'} \left\{ \frac{\sum_{i=1}^{q'} \lambda_i}{\sum_{i=1}^{n} \lambda_i} \geq 0.99 \right\}$ eigenvalues.
- There will be some leading eigenvectors in the feature space $\boldsymbol{T}$ that capture scene details rather than the real SPN components.
- This problem can not be simply solved by removing some specific eigenvectors from the feature space.

- Let us randomly select $M < q$ eigenvectors from the feature space $\boldsymbol{T}$ to form a random subspace $\boldsymbol{R}$. By repeating L times the process of randomly selecting subsets of $\boldsymbol{T}$, a set of random subspaces $\{\boldsymbol{R}_l \in \mathbb{R}^{N^2 \times M}\}_{l=1}^{L}$ is generated.
- An SPN template $\boldsymbol{x}$ is represented by a set of random features $\boldsymbol{y}_l = \boldsymbol{R}_l^\top \boldsymbol{x}, \quad l = 1, 2, \ldots, L.$
- To query whether an SPN sample is taken by a specific camera in the database:
  1. By performing the random subspace feature extraction on the training samples $\{\boldsymbol{x}_i\}_{i=1}^{n}$, a set of features $\{\boldsymbol{y}_{i,l}\}_{i=1}^{n}$ is generated in each random subspace $\boldsymbol{R}_l$.
  2. The reference $\overline{\boldsymbol{y}}_{j,l}$ of camera C_j in the subspace $\boldsymbol{R}_l$ is estimated

$$\overline{\boldsymbol{y}}_{j,l} = \frac{1}{E_j} \sum_{i=1}^{E_j} \boldsymbol{y}_{i,l}, \quad j = 1, 2, \ldots, c. \tag{5}$$

3. For a query SPN vector $\boldsymbol{x}_t$, we obtain a set of features $\{\boldsymbol{y}_{t,l}\}_{l=1}^{L}$. Once the query feature $\boldsymbol{y}_{t,l}$ and the camera reference SPN $\overline{\boldsymbol{y}}_{j,l}$ are generated, the camera identification problem is modeled as a binary hypothesis testing problem

$$\begin{aligned} \mathcal{H}_0 &: \boldsymbol{y}_{t,l} \neq \overline{\boldsymbol{y}}_{j,l} \quad \text{(the query image is not taken by the } j\text{th camera)} \\ \mathcal{H}_1 &: \boldsymbol{y}_{t,l} = \overline{\boldsymbol{y}}_{j,l} \quad \text{(the query image is taken by the } j\text{th camera)} \end{aligned} \tag{6}$$

The binary hypothesis testing problem (6) is solved by a correlation-based detector. That is, the normalized cross-correlation $\rho = \mathrm{corr}(\boldsymbol{y}_{t,l}, \overline{\boldsymbol{y}}_{j,l})$ is compared to a threshold η. The detector decides $\mathcal{H}_1$ if $\rho \geq \eta$ and $\mathcal{H}_0$ if $\rho < \eta$.

4. In each subspace $\boldsymbol{R}_l$ the aforementioned identification process is performed once. By repeating this process for each subspace, L decisions will be generated in total. The final decision is made according to the majority voting among the L decisions. If more than $L/2$ decisions are voted for $\mathcal{H}_1$, the final decision will assert that the query image is taken by the jth camera.

Source Camera Identification Experiments

Experimental results

- Experiments are conducted on the Dresden Image database[19].
- A total of 1200 images from 10 cameras are involved in the experiments, where each camera captured 120 images. These 10 camera devices belong to 4 brands, each brand has $2 \sim 3$ different models. For each camera, 20 images are used for training and the remaining 100 ones are used as query images for testing. There are 100×10 intraclass and 900×10 interclass correlation values in total.
- The noise residuals are computed from the luminance channel, as the luminance channel contains information of all the three channels. The experiments are performed on an image block of size 256×256 pixels cropped from the centre of a full size image.

[19] T. Gloe and R. Böhme, "The Dresden image database for benchmarking digital image forensics," *Journal Digital Forensic Practice*, vol. 3, no. 2-4, pp. 150–159, 2010.

Table 10: Cameras used in the experiments.

Camera	ID	Resolution
Canon Ixus70_A	C11	3072 × 2304
Canon Ixus70_B	C12	3072 × 2304
Canon Ixus70_C	C13	3072 × 2304
Nikon CoolPixS710_A	C21	4352 × 3264
Nikon CoolPixS710_B	C22	4352 × 3264
Samsung_L74wide_A	C31	3072 × 2304
Samsung_L74wide_B	C32	3072 × 2304
Samsung_L74wide_C	C33	3072 × 2304
Olympus_mju_1050SW_A	C41	3648 × 2736
Olympus_mju_1050SW_B	C42	3648 × 2736

Source Camera Identification Experiments

Experimental results

- There are only two parameters in the method, namely, the dimension of random subspace M and the number of random subspaces L.
- The performance of the method is improved by increasing the number of random subspaces. Since there is a trade off between the performance and the computational complexity, we set $L = 600$.
- The performance of the proposed method is same to that of the PCA-based extraction method when $M/q = 1$. As long as $M < q$, the proposed method achieves a higher true positive rate than the PCA-based extraction method.
- The performance of the method is not sensitive to L and M.
- The overall Receiver Operating Characteristic (ROC) curve is used to compare the performance of the different methods.

Source Camera Identification Experiments

Receiver Operating Characteristic curves

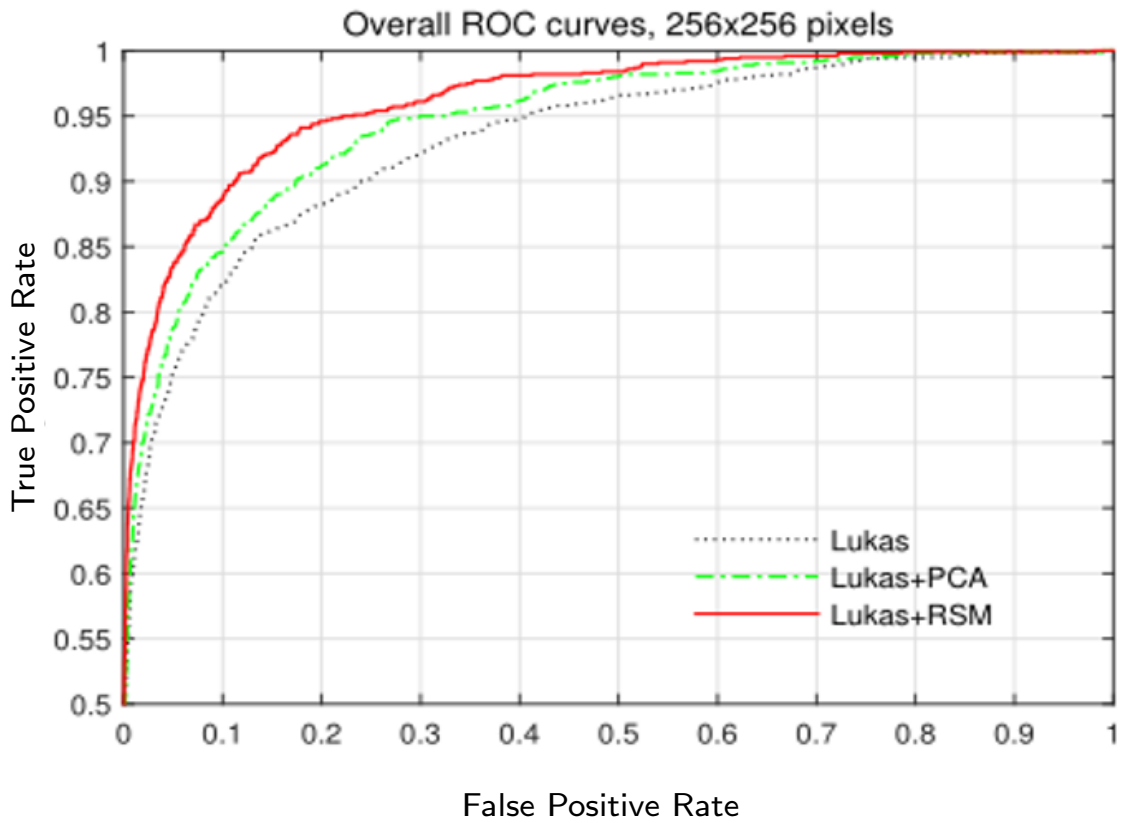


Figure 9: ROC curves of different variants of Lukas' method[20].

[20] J. Lukas, J. Fridrich, and M. Goljan, "Digital camera identification from sensor pattern noise," *IEEE Trans. Inf. Forensics Security*, vol. 1, no. 2, pp. 205–214, 2006.

Source Camera Identification Experiments

Receiver Operating Characteristic curves

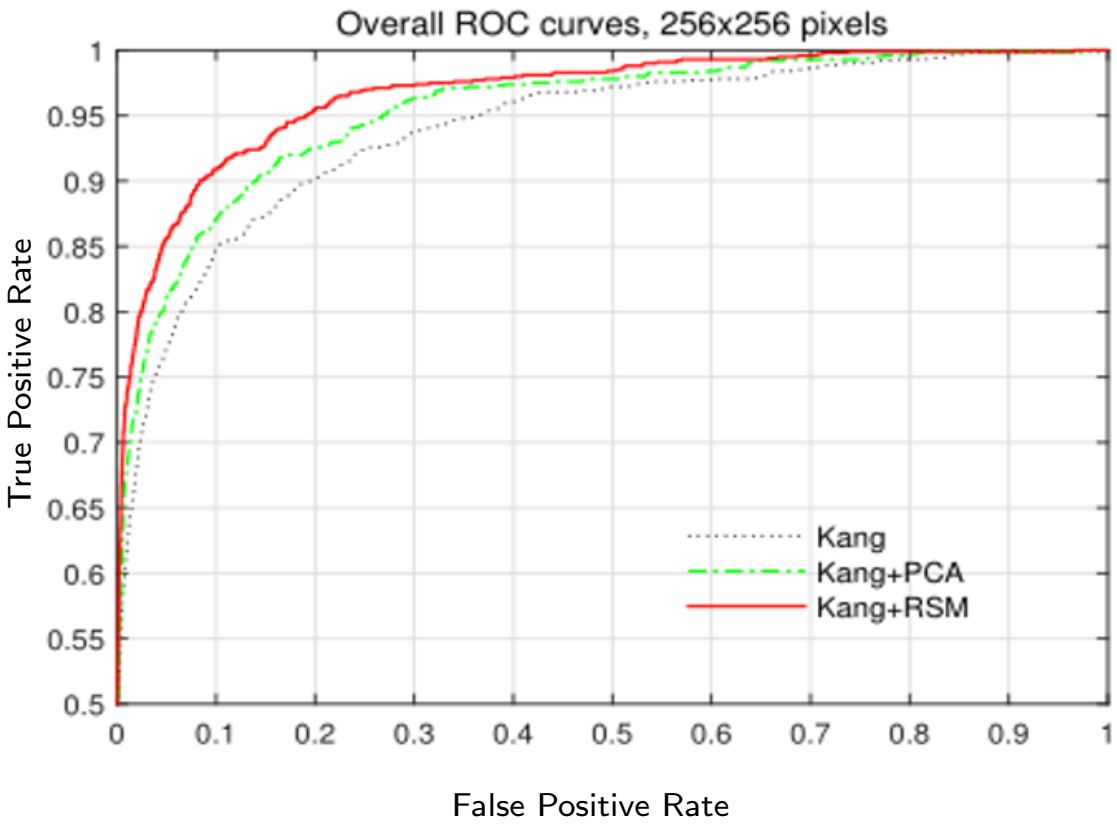


Figure 10: ROC curves of different variants of Kang's method[21].

[21] X. Kang, J. Chen, K. Lin, and A. Peng, "A context-adaptive SPN predictor for trustworthy source camera identification," *EURASIP Journal Image and Video Processing*, vol. 2014, no. 1, pp. 1–11, 2014.

Explainable Closed-Set Source Device Identification Using Log-Mel Spectrograms from Videos' Audio

Contributions

- An innovative method for Source Device Identification (SDI) has been developed[22].
- Deployment of an optimized ResNet-based model[23] employing Neural Architecture Search (NAS) for classifying camera devices as well as sets of camera devices of the same model.
- Gradient-weighted Class Activation Mapping[24] is integrated to identify and emphasize key features within log-Mel spectrograms extracted from the video's audio track.
- Bandpass filtering is applied to selectively preserve mid to high-frequency information.

[22] C. Korgialas, G. Tzolopoulos, and C. Kotropoulos, "On Explainable Closed-Set Source Device Identification Using log-Mel Spectrograms from Videos' Audio: A Grad-CAM Approach," *IEEE Access*, vol. 12, pp. 121822-121836, 2024.

[23] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Computer Vision Pattern Recognition (CVPR)*, June 2016, pp. 770–778.

[24] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-CAM: Visual explanations from deep networks via gradient-based localization," in *Proc. IEEE Int. Conf. Computer Vision (ICCV)*, October 2017, pp. 618–626.

Explainable Closed-Set Source Device Identification Using Log-Mel Spectrograms from Videos' Audio

VISION dataset

- The VISION dataset[25] is utilized, comprising images and videos captured across various scenes and imaging conditions.
  - 35 camera devices, representing 29 camera models and 11 camera brands, are encompassed within VISION.
- VISION includes 648 native videos, which remain unaltered post-capture by the camera.
  - These native videos were disseminated via social media platforms like YouTube and WhatsApp, with corresponding versions available in the dataset.
- The dataset is partitioned into training, testing, and validation sets to conduct a typical five-fold stratified cross-validation.

[25] D. Shullani, M. Fontani, M. Juliani, O. A. Shaya, and A. Piva, "VISION: A video and image dataset for source identification," *EURASIP Journal on Information Security*, (2017), pp. 1–16.

Explainable Closed-Set Source Device Identification Using Log-Mel Spectrograms from Videos' Audio

VISION dataset

Table 11: The 35 devices featured in the VISION dataset.

Device ID	Brand & Model	Device ID	Brand & Model
D01	Samsung Galaxy S3 Mini	D19	Apple iPhone 6 Plus
D02	Apple iPhone 4s	D20	Apple iPad Mini
D03	Huawei P9	D21	Wiko Ridge 4G
D04	LG D290	D22	Samsung Galaxy Trend Plus
D05	Apple iPhone 5c	D23	Asus Zenfone 2 Laser
D06	Apple iPhone 6	D24	Xiaomi Redmi Note 3
D07	Lenovo P70A	D25	OnePlus A3000
D08	Samsung Galaxy Tab 3	D26	Samsung Galaxy S3 Mini
D09	Apple iPhone 4	D27	Samsung Galaxy S5
D10	Apple iPhone 4s	D28	Huawei P8
D11	Samsung Galaxy S3	D29	Apple iPhone 5
D12	Sony Xperia Z1 Compact	D30	Huawei Honor 5c
D13	Apple iPad 2	D31	Samsung Galaxy S4 Mini
D14	Apple iPhone 5c	D32	OnePlus A3003
D15	Apple iPhone 6	D33	Huawei Ascend
D16	Huawei P9 Lite	D34	Apple iPhone 5
D17	Microsoft Lumia 640 LTE	D35	Samsung Galaxy Tab A
D18	Apple iPhone 5c		

Explainable Closed-Set Source Device Identification Using Log-Mel Spectrograms from Videos' Audio

Log-mel spectrogram generation

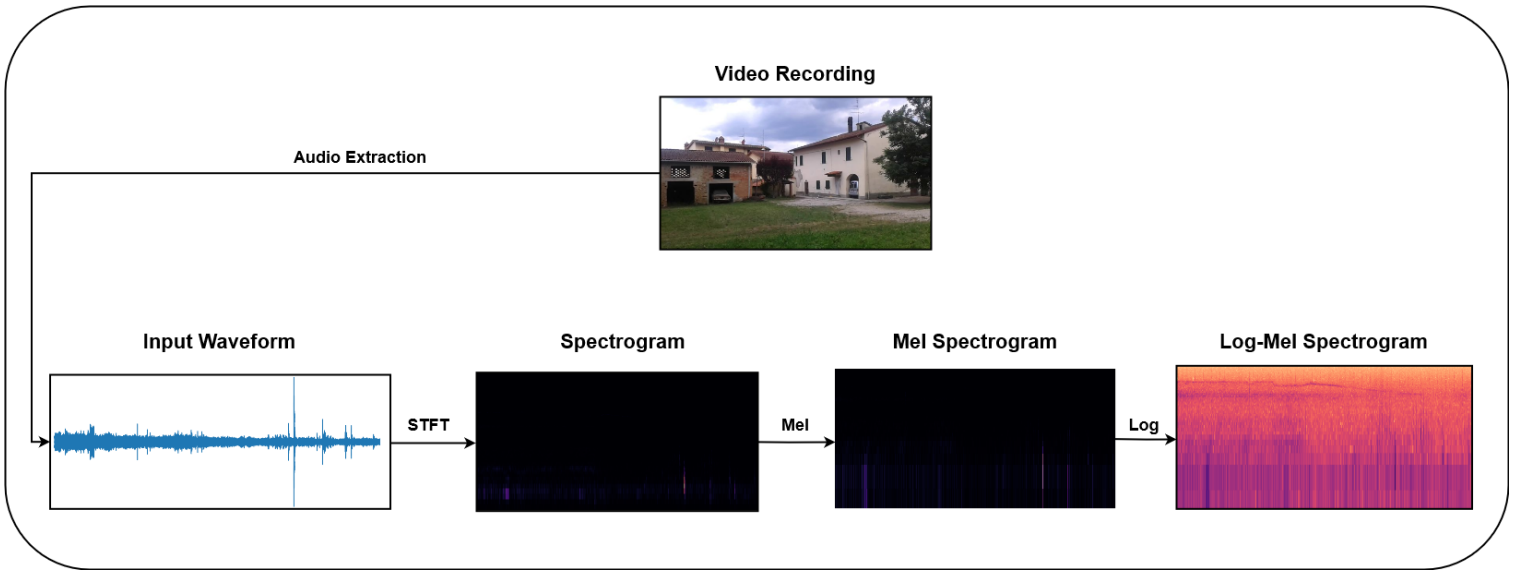


Figure 11: Transformation from the raw waveform of the audio signal to the log-Mel spectrogram.

- Grad-CAM identifies the feature maps A^k within a particular convolutional layer. Using backpropagation, A^k are associated with a weight α_k^c calculated by pooling the gradients of the class score, Y^c, with respect to the feature map. The pooling operation is a Global Average Pooling (GAP) that summarizes the gradients across the spatial dimensions $u \times v$ of the feature map.
- The specific weight for the k-th feature map and class c is computed as:

$$\alpha_k^c = \frac{1}{Z} \sum_{i,j} \frac{\partial Y^c}{\partial A_{i,j}^k}, \tag{7}$$

where Z is the number of pixels in the feature map. The indices i and j stand for the row and column of the feature map, respectively.

- The weights α_k^c are used in a linear combination with the feature maps to produce a localization map for class c, which is processed through a Rectified Linear Unit (ReLU). The ReLU function is applied to ensure that only features with a positive influence on the class score are considered, setting all negative values to zero.

$$L^c = \text{ReLU}\left(\sum_k \alpha_k^c \, A^k\right). \tag{8}$$

- The localization map L^c is then upscaled to the dimensions of the input image through bilinear interpolation, resulting in the final visual explanation that highlights the influential regions for the specific class c.

Explainable Closed-Set Source Device Identification Using Log-Mel Spectrograms from Videos' Audio

Grad-CAM

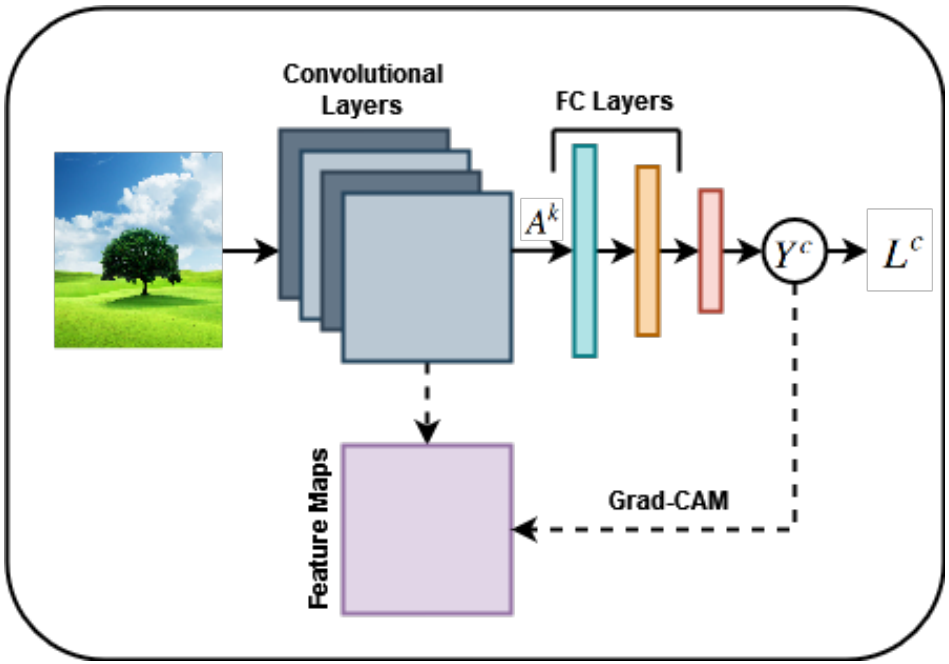


Figure 12: Flowchart of the Grad-CAM approach for visualizing and explaining CNN features.

Explainable Closed-Set Source Device Identification Using Log-Mel Spectrograms from Videos' Audio

Flowchart

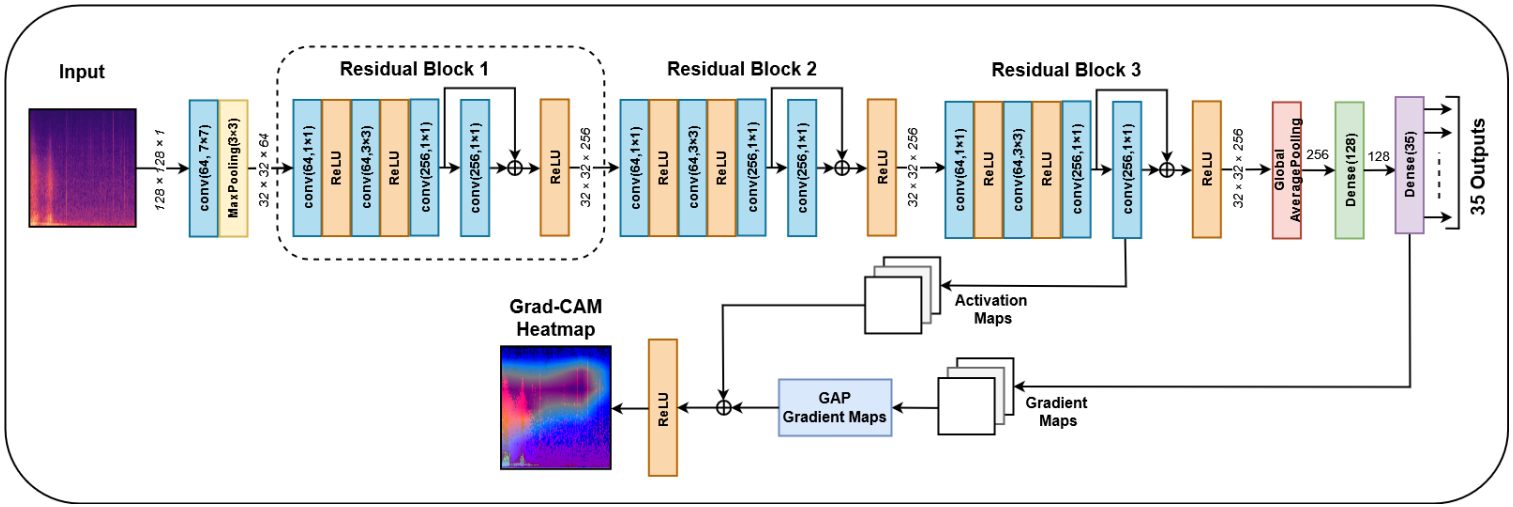


Figure 13: Flowchart depicting the ResNet-based CNN architecture for SDI with Grad-CAM visualizations highlighting influential regions in audio spectrograms.

Explainable Closed-Set Source Device Identification Using Log-Mel Spectrograms from Videos' Audio

Log-Mel spectrograms overlayed with Grad-CAM visual explanations

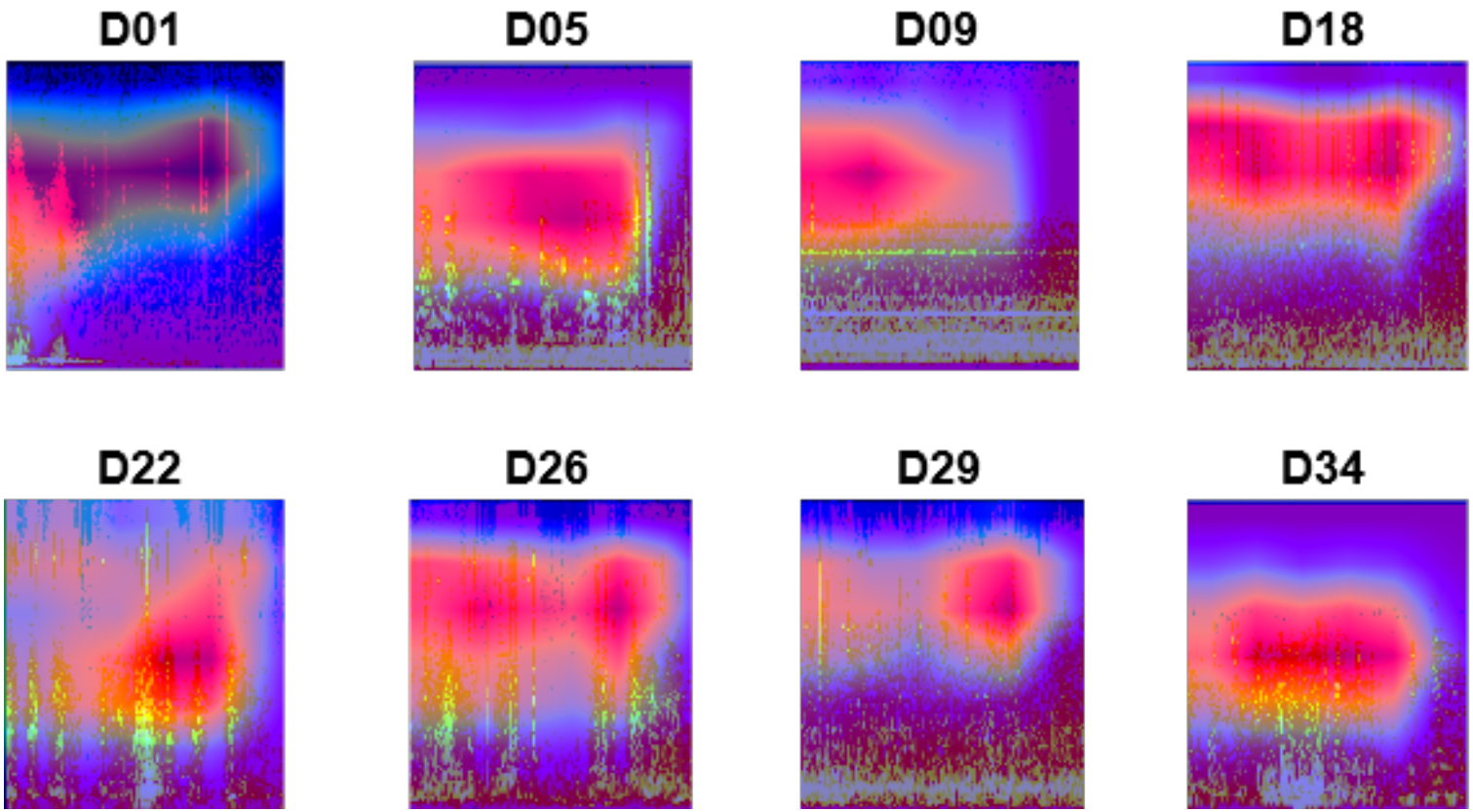


Figure 14: Eight randomly selected log-Mel spectrograms overlayed with Grad-CAM visual explanations, highlight the key areas of interest for camera device classification within the ResNet-based model.

Explainable Closed-Set Source Device Identification Using Log-Mel Spectrograms from Videos' Audio

Confusion matrices using the overall test data

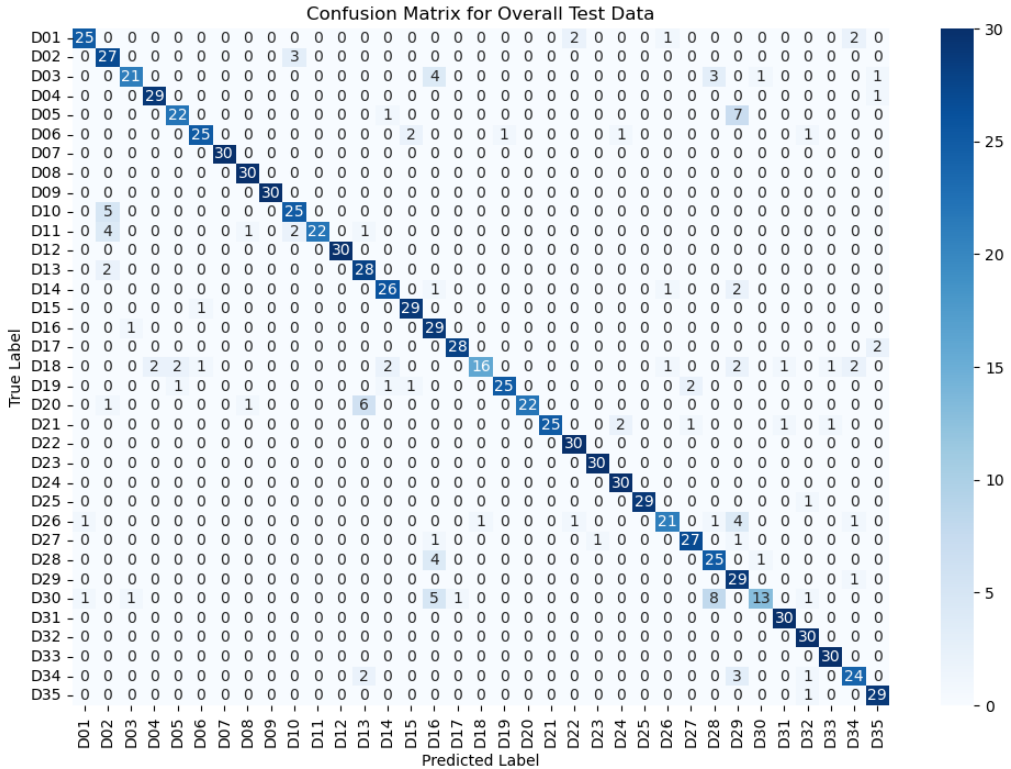


Figure 15: Confusion matrix obtained by evaluating the ResNet-based model on the VISION test dataset using log-Mel spectrograms of video audio from the test set along the 35 devices.

Explainable Closed-Set Source Device Identification Using Log-Mel Spectrograms from Videos' Audio

Confusion matrices using the overall test data

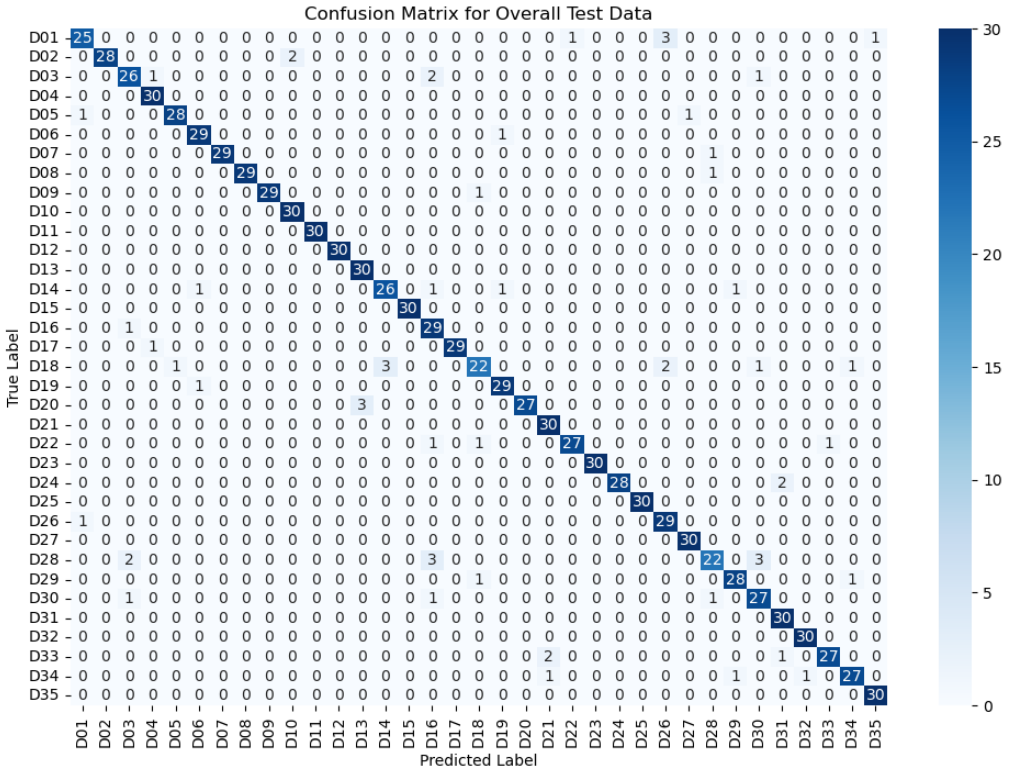


Figure 16: Confusion matrix obtained by evaluating the Grad-CAM-driven ResNet-based model on the VISION test dataset using band-passed filtered log-Mel spectrograms of video audio from the test set along the 35 devices.

- A bootstrap methodology[26] is used to compare the performance of the ResNet-based model and Grad-CAM-driven ResNet-based model across the overall, flat, indoor, and outdoor test data.
- Accuracy of model m, $AC(m)$:

$$AC(m) = \frac{\sum_{i=1}^{M} CM(m)_{ii}}{\sum_{i=1}^{M}\sum_{j=1}^{M} CM(m)_{ij}}, \tag{9}$$

where $CM(m)_{ii}$ are the diagonal elements of the confusion matrix and $CM(m)_{ij}$ are the off-diagonal elements, with M being the number of classes (i.e., device models).

[26] P. J. Hardin and J. M. Shumway, "Statistical significance and normalized confusion matrices," *Photogramm. Engineer. Remote Sens.*, vol. 63, no. 6, pp. 735–739, Jan. 1997.

- Differences between the accuracies of the two models are subjected to a bootstrapping process to generate empirical distributions under the null hypothesis $\mathcal{H}_0$, which posits no significant difference in model performance. These differences, D_i, are determined for each bootstrap sample and are described by the following expression:

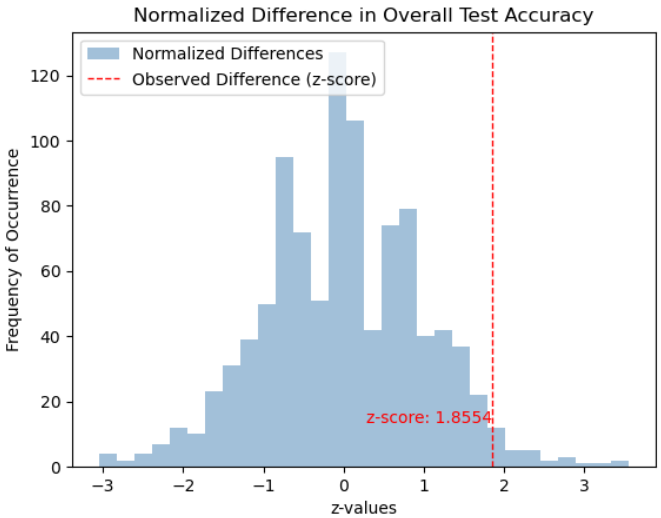
$$D_i = AC(m_2^{(i)}) - AC(m_1^{(i)}), \tag{10}$$

where $m_1^{(i)}$ and $m_2^{(i)}$ are the i-th bootstrap samples for the first and second models, respectively.

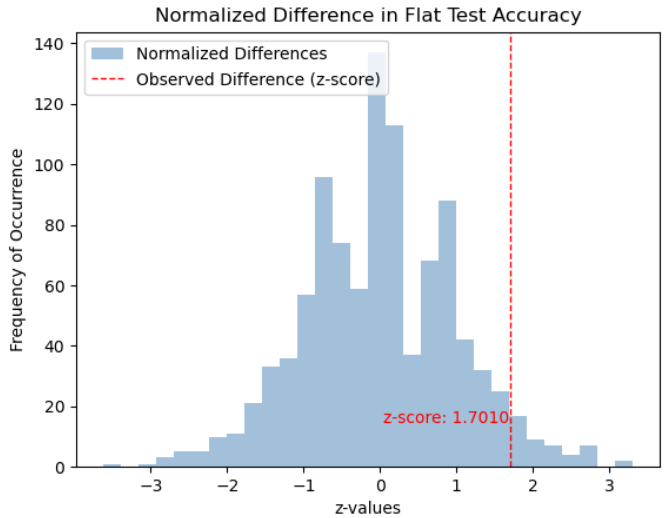
- The z-scores for each observed difference D_{obs} in accuracy between models are indicated by dashed red lines. The observed z-scores for overall, flat, indoor, and outdoor test accuracy deviations are 1.854, 1.701, 1.4104, and 2.43, respectively. The outdoor z-score notably surpasses the 1.96 threshold, indicating a significant performance difference from the expected mean under $\mathcal{H}_0$ at significance level 0.05.

Explainable Closed-Set Source Device Identification Using Log-Mel Spectrograms from Videos' Audio

Bootstrap analysis



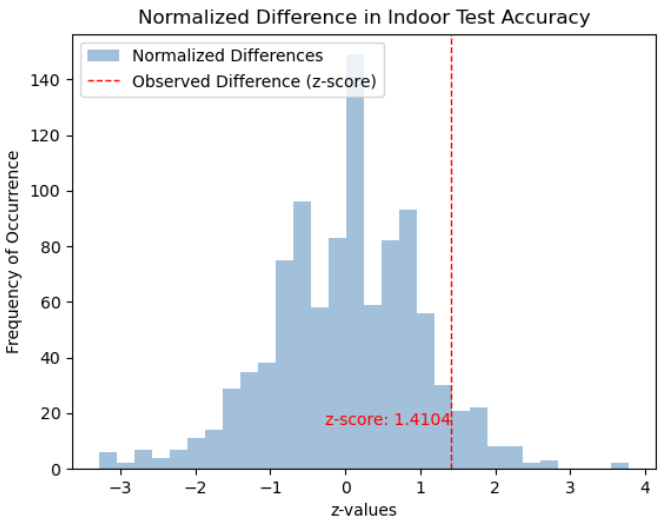
(a) Histogram of bootstrapped overall test accuracy differences.



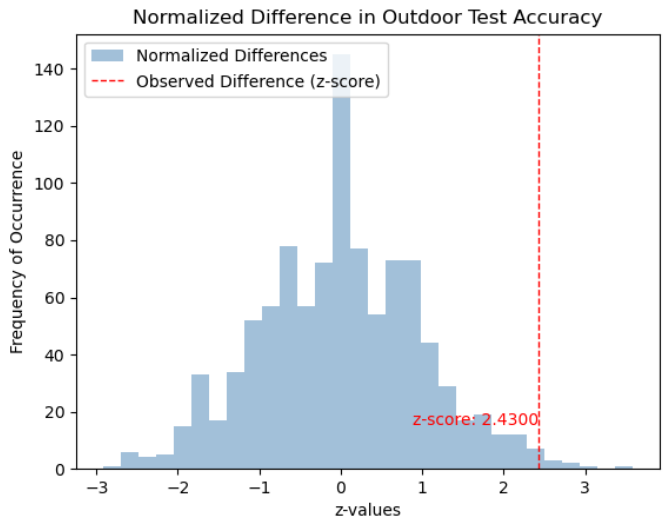
(b) Histogram of bootstrapped flat test accuracy differences.

Explainable Closed-Set Source Device Identification Using Log-Mel Spectrograms from Videos' Audio

Bootstrap analysis



(c) Histogram of bootstrapped indoor test accuracy differences.



(d) Histogram of bootstrapped outdoor test accuracy differences.

Figure 17: Comparative analysis of Grad-CAM-driven ResNet-based model test accuracy with histograms displaying the normalized distribution of bootstrapped accuracy differences (z-values) for overall, flat, indoor, and outdoor test datasets. The dashed red line in the histograms marks the observed discrepancy in accuracy.

- The p-values are also computed to evaluate the statistical significance of the observed differences in model accuracy. This interpretation is guided by the computation of the p-value, which is the proportion of the bootstrapped differences D_i that are as extreme or more than the observed difference D_{obs}, i.e.,

$$p\text{-value} = \frac{1}{N} \sum_{i=1}^{N} \mathbb{I}(|D_i| \geq |D_{obs}|), \tag{11}$$

where $\mathbb{I}(\cdot)$ is the indicator function that equals 1 if the condition is met and 0 otherwise, and $N = 1000$ is the total number of bootstrap iterations. The observed difference D_{obs} is defined as the discrepancy in accuracy between the two models under comparison.

Explainable Closed-Set Source Device Identification Using Log-Mel Spectrograms from Videos' Audio

Bootstrap analysis

- For the overall test data, the observed difference has a p-value of 0.054, suggesting marginally significant evidence against $\mathcal{H}_0$. For the flat test scenario, a p-value of 0.089 suggests marginally significant evidence against $\mathcal{H}_0$, while the indoor test scenario, with a p-value of 0.142, exhibits non-significant evidence against $\mathcal{H}_0$. The outdoor test scenario shows a p-value of 0.015, suggesting significant evidence against $\mathcal{H}_0$.

Explainable Closed-Set Source Device Identification Using Log-Mel Spectrograms from Videos' Audio

Comparison with the state-of-the-art

- The Grad-CAM-driven ResNet-based model is assessed by adopting the dataset-splitting approach in[27].
- The VGG-based model is the sole method employing log-Mel spectrograms, achieving an accuracy of 65.78% across the overall test data. In contrast, the proposed method demonstrates a significant improvement, achieving a test accuracy of 97.07% on the same dataset. The superiority of our approach is further corroborated by the confusion matrix presented in Figure 18, which visually articulates the precision of the proposed model.
- However, a fusion framework is introduced to further enhance the CMI accuracy[27], leveraging both video frames and audio log-Mel spectrograms, resulting in an accuracy of 99% for the overall test data.

Explainable Closed-Set Source Device Identification Using Log-Mel Spectrograms from Videos' Audio

Comparison with the state-of-the-art

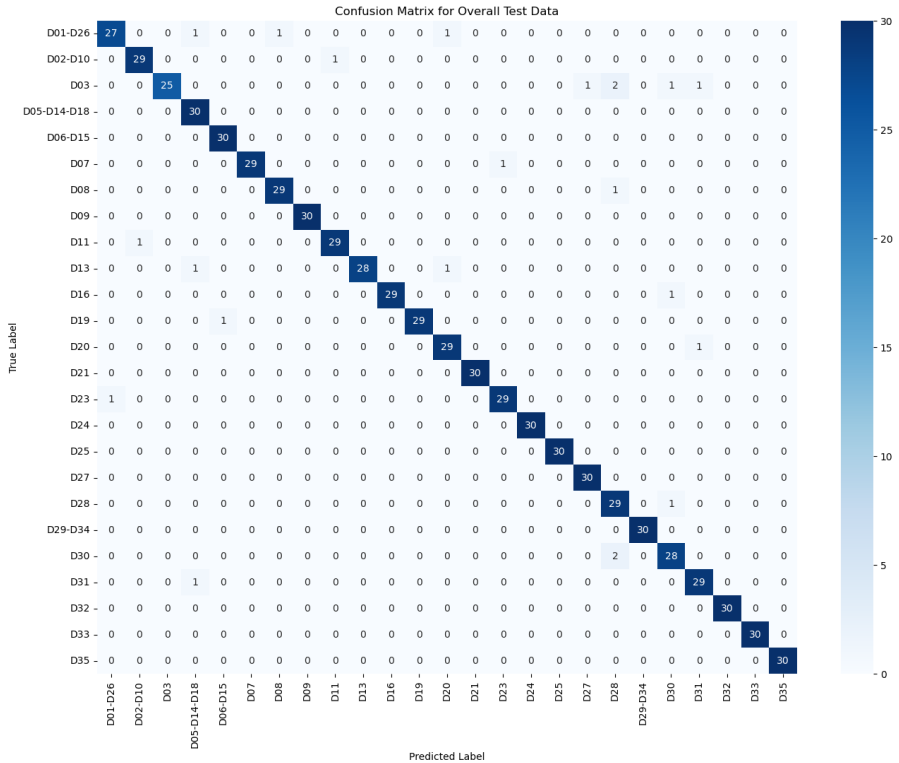


Figure 18: The confusion matrix obtained by evaluating the Grad-CAM-driven ResNet-based model on the VISION dataset for the overall test data following the setup in[27].

[27] D. Dal Cortivo, S. Mandelli, P. Bestagini, and S. Tubaro, "CNN-based multi-modal camera model identification on video sequences," *J. Imag.*, vol. 7, no. 8, p. 135, Aug. 2021.

Attention-Based Source Device Identification Using Audio Content from Videos and Grad-CAM Explanations

Contributions

- A Convolutional Block Attention Module (CBAM)[28] is integrated into a ResNet-based model, with hyperparameters optimized using NAS[29], to enhance the SDI accuracy among the 35 devices of the VISION dataset.
- The proposed framework leverages spatial and temporal modules to improve the model's ability to focus on specific areas of log-Mel spectrograms derived from the audio content of videos, as highlighted by Grad-CAM explanations.

[28] S. Woo, J. Park, J.-Y. Lee, and I. Kweon, "CBAM: Convolutional block attention module," in *Proc. Europ. Conf. Comput. Vision*, Sep. 2018, pp. 3–19.

[29] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, "Optuna: A next-generation hyperparameter optimization framework," in *Proc. Int. Conf. Knowledge Discov. Data Mining*, Jul. 2019, pp. 2623–2631.

Attention-Based Source Device Identification Using Audio Content from Videos and Grad-CAM Explanations

CBAM-ResNet-based model

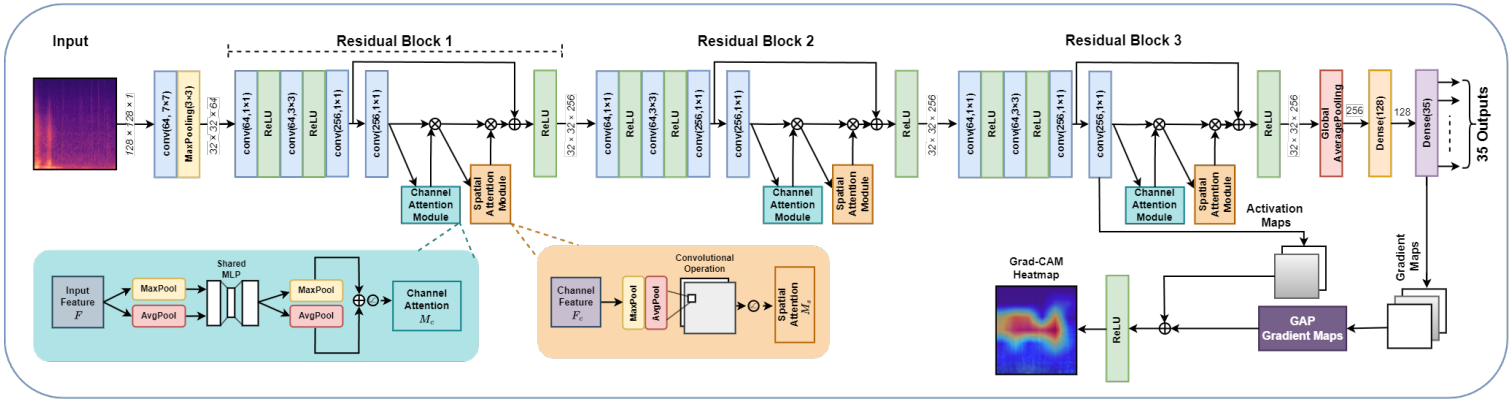


Figure 19: Flowchart depicting the CBAM-ResNet-based model for SDI, illustrating the integration of CBAM with the ResNet architecture. Grad-CAM explanations are derived from the final residual block of the network.

Attention-Based Source Device Identification Using Audio Content from Videos and Grad-CAM Explanations

CBAM-ResNet-based model

Table 12: Comparison of validation accuracy (%) and average AUC across all classes between the proposed CBAM-ResNet-based model and the approaches developed in[22].

Model	Preprocessing Method	Overall		Flat		Indoor		Outdoor	
		Accuracy (%)	AUC	Accuracy (%)	AUC	Accuracy (%)	AUC	Accuracy (%)	AUC
ResNet-based[22]	No filtering	89.90	0.975	89.71	0.975	92.29	0.979	87.71	0.966
Grad-CAM-driven ResNet-based[22]	Bandpass filtering	94.76	0.984	96.29	0.987	94.57	0.988	93.43	0.979
CBAM-ResNet-based	No filtering	**97.81**	**0.989**	**98.29**	**0.990**	**98.86**	**0.990**	**97.14**	**0.982**

Attention-Based Source Device Identification Using Audio Content from Videos and Grad-CAM Explanations

CBAM-ResNet-based model

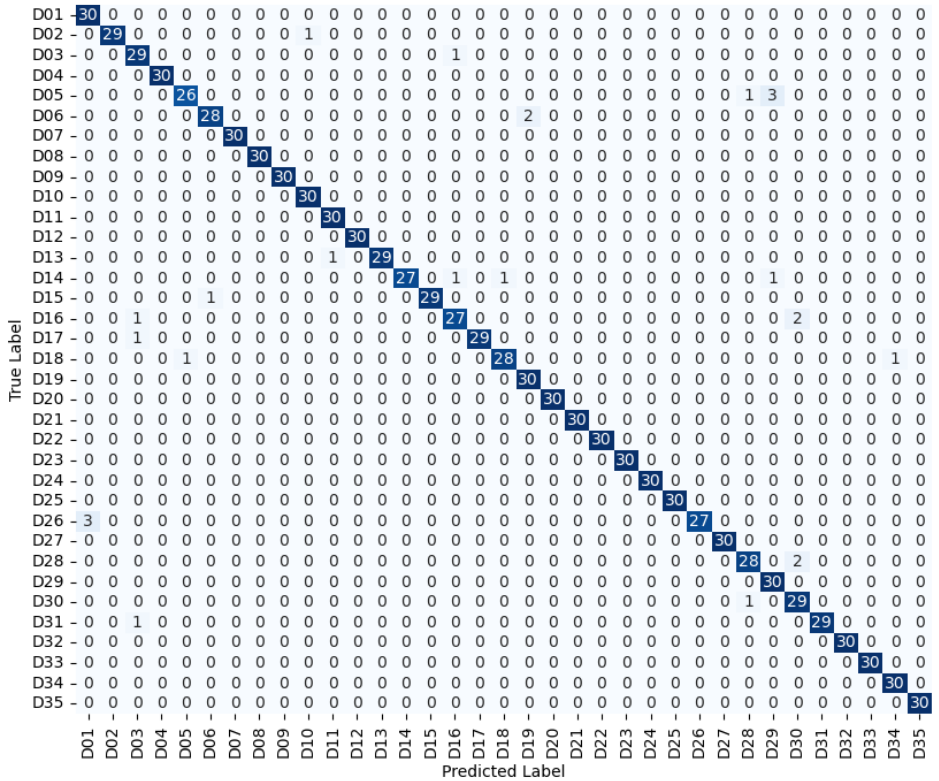


Figure 20: Confusion matrix obtained by evaluating the CBAM-ResNet-based model on the VISION test dataset.

Attention-Based Source Device Identification Using Audio Content from Videos and Grad-CAM Explanations

CBAM-ResNet-based model

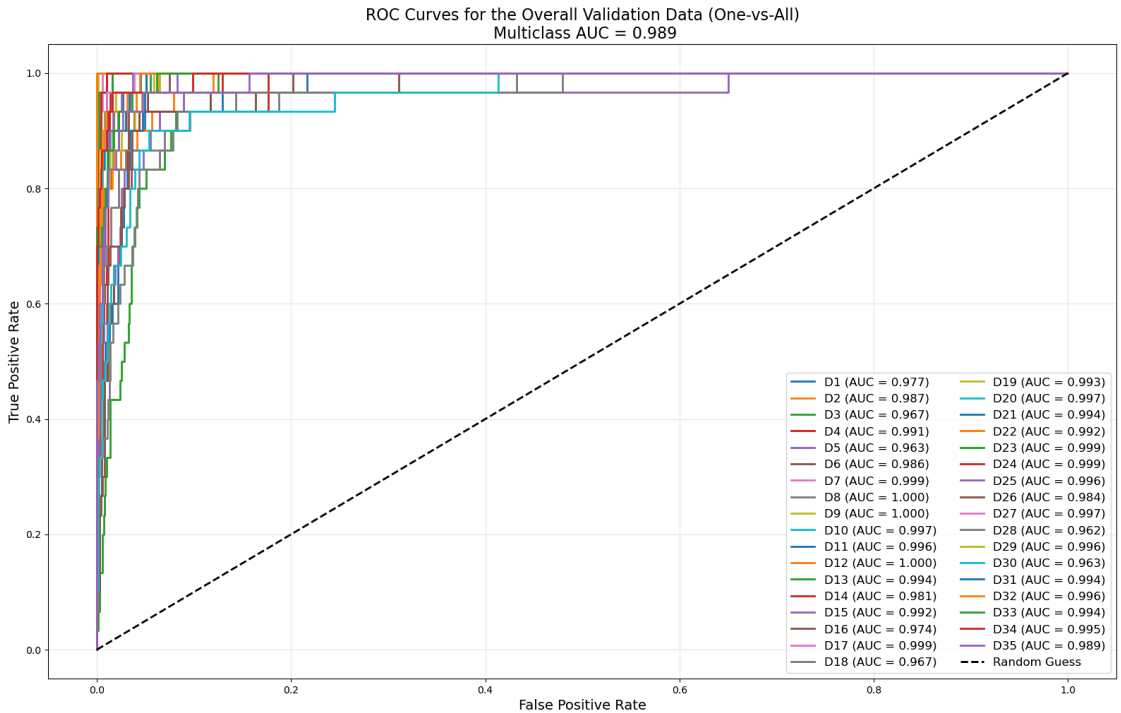


Figure 21: ROC curves generated for the validation set illustrate the overall data scenario across 35 devices in the VISION dataset.

Attention-Based Source Device Identification Using Audio Content from Videos and Grad-CAM Explanations

CBAM-ResNet-based model

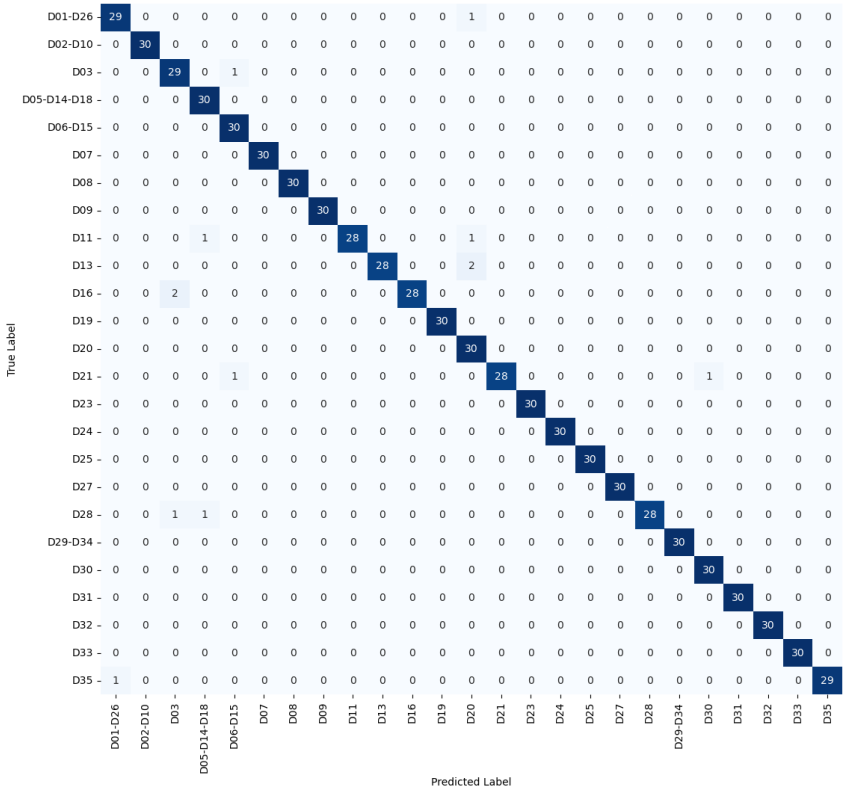


Figure 22: Comparison of confusion matrices for the overall test set, following the methodology in[27].

Camera Model Identification Using Audio and Visual Content

Contributions

- A framework for Camera Model Identification (CMI) is developed, treating CMI as a classification problem[30] .
- CNNs trained on either audio or visual content are employed for this purpose. Experimental findings showcase promising performance when employing either audio or visual content individually.
- Additionally, based on the classification decisions derived from the audio and visual content, late fusion is applied to these decisions using fundamental fusion rules, specifically the product and sum rules[31].

[30] I. Tsingalis, C. Korgialas, and C. Kotropoulos, "Camera Model Identification Using Audio and Visual Content from Videos," in *Proc. 2024 IARIA Annual Congress on Frontiers in Science, Technology, Services, and Applications*, Porto, Portugal, June 30 – July 4, 2024.

[31] J. Kittler, M. Hatef, R. P. Duin, and J. Matas, "On combining classifiers," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 3, pp. 226–239, 1998.

- *Audio content*: Three windows and hop parameters are used to construct a three-channel Log Mel-Spectogram for each audio recording.
- *Visual content*: The raw video frames are used.

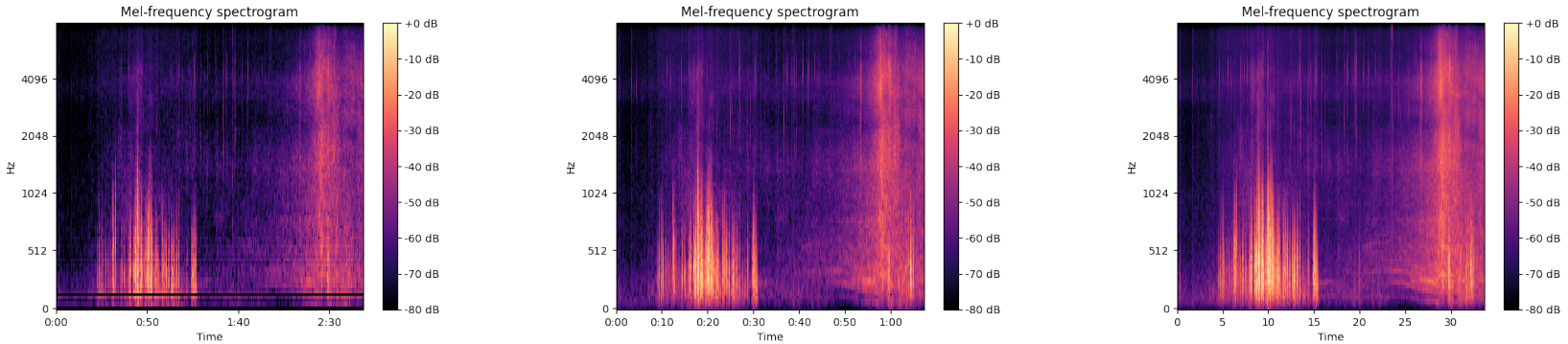


Figure 23: Three Mel-frequency spectrograms.

- We denote the instances of the mth modality as $\{\boldsymbol{\gamma}_m^{(n)}\}_{n=1}^{N}$.
- Let $\boldsymbol{W}_m^{[l]}$ and $f_{\boldsymbol{W}_m^{[l]}}^{[l]}$ be the parameters and the activation function of the lth layer on a Neural Network (NN) related to the mth modality, with $l = 1, \ldots, L-1$.
- Let $\{\boldsymbol{w}_m^{c',[L]}\}_{c'=1}^{C}$ be the collection of parameters belonging to the Lth layer where $\boldsymbol{w}_m^{c',[L]}$ is associated with the c'th classification node, with $c' = 1, \ldots, C$.

- Let

$$\Pr(\mathcal{C}_{c'} \mid \gamma_m^{(n)} ; \boldsymbol{w}_m^{c',[L]}) = \frac{\exp(\boldsymbol{w}_m^{c',[L]^\top} \boldsymbol{a}_m^{(n)[L-1]})}{\sum_{c=1}^{C} \exp(\boldsymbol{w}_m^{c,[L]^\top} \boldsymbol{a}_m^{(n)[L-1]})} \tag{12}$$

be the classification probabilities of the c' classification node where

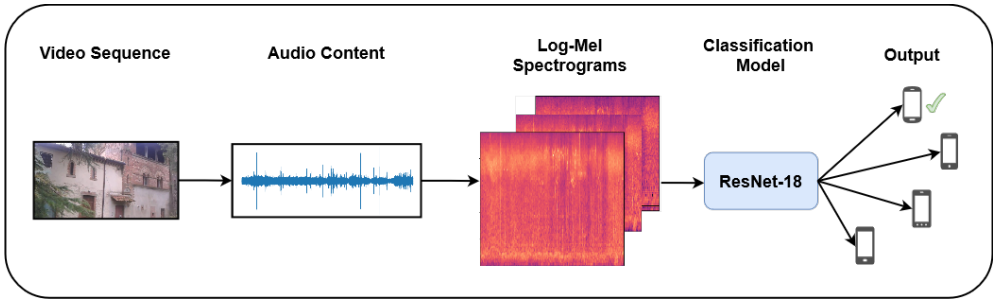
$$\boldsymbol{a}_m^{(n)[L-1]} = \left(f_{\boldsymbol{W}_m^{[L-1]}}^{[L-1]} \circ \cdots \circ f_{\boldsymbol{W}_m^{[1]}}^{[1]} \right) (\gamma_m^{(n)}) \tag{13}$$

is the activation output of the penultimate CNN layer, and $\circ$ denotes function composition.

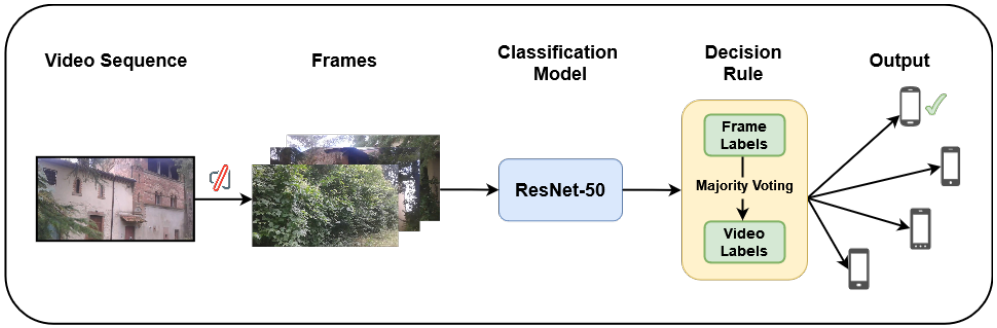
- **Audio and Visual Content NN Classifier:**
  - Audio Content NN Classifier: `ResNet18`
  - Visual Content NN Classifier: `ResNet50`

Camera Model Identification Using Audio and Visual Content

Classification pipeline



(a) Audio classification pipeline using `ResNet18`.



(b) Video frame classification pipeline using `ResNet50`.

Figure 24: Flowchart depicting the CMI employing frames extracted from video sequences.

In addition, the classification probabilities of the nth sample $\boldsymbol{\gamma}_m^{(n)}$ related to the mth modality are given by

$$\boldsymbol{p}_m^{(n)[L]} = \begin{bmatrix} \Pr(\mathcal{C}_1 \mid \boldsymbol{\gamma}_m^{(n)}; \boldsymbol{w}_m^{1,[L]}) \\ \Pr(\mathcal{C}_2 \mid \boldsymbol{\gamma}_m^{(n)}; \boldsymbol{w}_m^{2,[L]}) \\ \vdots \\ \Pr(\mathcal{C}_C \mid \boldsymbol{\gamma}_m^{(n)}; \boldsymbol{w}_m^{C,[L]}) \end{bmatrix} \in \mathbb{R}^C. \tag{14}$$

In the following, the superscript $[L]$ is omitted for simplicity. Given the samples $\{\boldsymbol{\gamma}_m^{(n)}\}_{n=1}^N$ of the mth modality, we obtain

$$\boldsymbol{P}_m = [\boldsymbol{p}_m^{(1)}, \boldsymbol{p}_m^{(2)}, \ldots, \boldsymbol{p}_m^{(N)}] \in \mathbb{R}^{C \times N}. \tag{15}$$

- **Unimodal Testing Procedure.** The predicted classes of each sample $\{\boldsymbol{\gamma}_m^{(n)}\}_{n=1}^{N}$ are given by

$$\boldsymbol{c}_m = [\mathcal{C}_m^1, \mathcal{C}_m^2, \ldots, \mathcal{C}_m^N]^\top \in \mathbb{R}^N, \tag{16}$$

where

$$\mathcal{C}_m^n = \underset{c=1,\ldots,C}{\operatorname{argmax}} \left[\boldsymbol{p}_m^{(n)}\right]_c, \tag{17}$$

is the predicted class of the nth sample with $\mathcal{C}_m^n \in \{\mathcal{C}_{c=1}^C\}$.

- **Multimodal Testing Procedure.** The *product rule* is given by

$$\boldsymbol{P}_{\text{prod}} = \boldsymbol{P}_1 \odot \boldsymbol{P}_2 \odot \cdots \odot \boldsymbol{P}_M \in \mathbb{R}^{C \times N}, \tag{18}$$

where $\odot$ denotes the Hadamard, element-wise, product. The *sum rule* is given by

$$\boldsymbol{P}_{\text{sum}} = \boldsymbol{P}_1 + \boldsymbol{P}_2 + \cdots + \boldsymbol{P}_M \in \mathbb{R}^{C \times N}. \tag{19}$$

Camera Model Identification Using Audio and Visual Content

Unimodal accuracy

Table 13: Accuracy (%) using either visual or audio content

	Visual-ResNet-50			**Audio-ResNet-18**		
	Native	WhatsApp	YouTube	Native	WhatsApp	YouTube
Fold 0	88.31	67.53	77.02	96.10	93.50	91.9
Fold 1	85.70	83.11	72.97	94.80	90.90	93.24
Fold 2	89.60	63.63	77.02	90.90	88.31	95.94
Fold 3	89.47	68.42	63.51	93.42	94.73	82.43
Fold 4	88.15	64.47	78.37	94.73	88.15	95.94
Mean ± StD	88.24 ± 1.4	69.43 ± 7.07	71.77 ± 5.44	93.99 ± 1.76	91.11 ± 2.66	91.89 ± 4.98

Camera Model Identification Using Audio and Visual Content

Late fusion accuracy

Table 14: Accuracy (%) using the product and sum rule

	Product Rule			**Sum Rule**		
	Native	WhatsApp	YouTube	Native	WhatsApp	YouTube
Fold 0	97.40	94.80	95.94	97.40	96.10	94.59
Fold 1	97.40	94.80	94.59	96.10	96.10	93.24
Fold 2	98.70	93.50	95.94	97.40	90.90	97.29
Fold 3	97.36	94.73	90.54	94.73	97.36	86.48
Fold 4	97.36	86.84	95.94	96.05	88.15	97.29
Mean ± StD	97.64 ± 0.52	92.93 ± 3.08	95.59 ± 0.52	96.33 ± 0.99	93.72 ± 3.56	93.77 ± 3.97

Next, we study three null hypotheses:

- $\mathcal{H}_{0,1}$: The classification performances achieved by the two fusion rules are equivalent.
- $\mathcal{H}_{0,2}$: The classification performance achieved solely with visual content is equivalent to that achieved with the product rule.
- $\mathcal{H}_{0,3}$: The classification performance achieved solely with audio content is equivalent to that achieved with the product rule.

Camera Model Identification Using Audio and Visual Content

Statistical significance

- The p-values are computed by applying McNemar's significance test[32].
- Most of the p-values exceed the predetermined significance threshold, so we lack significant evidence against $H_{0,1}$.
- We have significant evidence against $\mathcal{H}_{0,2}$.
- For $\mathcal{H}_{0,3}$, most of the p-values exceed the predetermined significance threshold, so we lack significant evidence against $\mathcal{H}_{0,3}$.

Table 15: McNemar's p-values to evaluate the null hypotheses $H_{0,2}$ and $H_{0,3}$

	Visual-ResNet-50			**Audio-ResNet-18**		
	Native	WhatsApp	YouTube	Native	WhatsApp	YouTube
Fold 0	0.023	10^{-5}	0.001	1.0	1.0	0.371
Fold 1	0.007	0.026	0.001	0.617	0.248	1.0
Fold 2	0.044	10^{-5}	0.001	0.041	0.133	0.479
Fold 3	0.041	10^{-5}	10^{-5}	0.248	0.617	0.007
Fold 4	0.045	10^{-4}	0.002	0.617	1.0	0.479

[32] Q. McNemar, "Note on the sampling error of the difference between correlated proportions or percentages," *Psychometrika*, vol. 12, no. 2, pp. 153–157, 1947.

Explainable Camera Model Identification

Contributions

- Grad-CAM explanations are incorporated into the audio component of the framework, improving the interpretability of the audio-based decisions by highlighting important regions in the log-Mel spectrograms[33].
- A band-pass filtering method is introduced based on Grad-CAM explanations, addressing the `ResNet-18` model's tendency to focus on specific frequency regions.

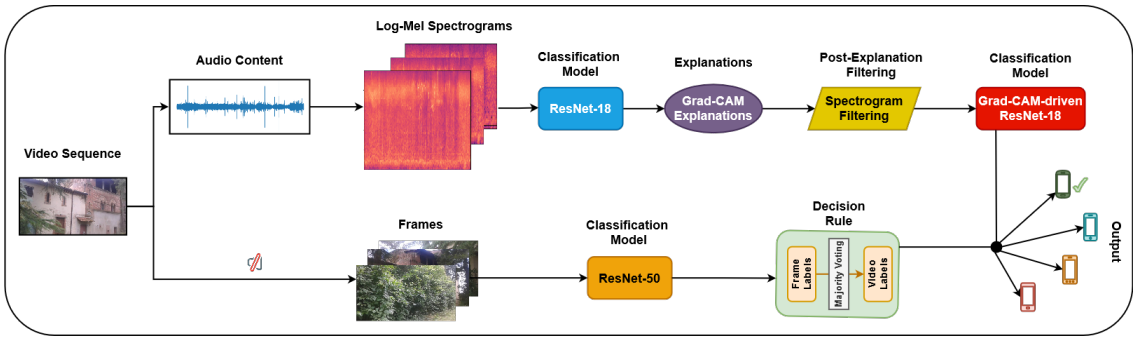


Figure 25: Flowchart depicting the explainable CMI using the audio (top) and video (bottom) content.

[33] C. Korgialas, I. Tsingalis, and C. Kotropoulos, "Explainable Camera Model Identification Employing log-Mel Spectrograms from Videos' Audio," in *Proc. IEEE Asilomar Conf. on Signals, Systems, and Computers*, Monterey CA, October 27-30, 2024.

Explainable Camera Model Identification

Grad-CAM heatmaps

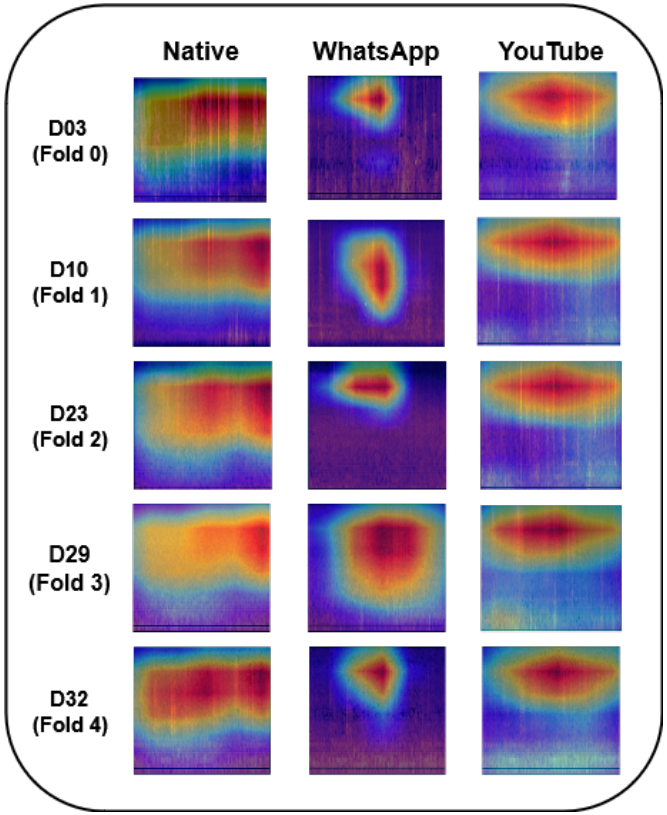


Figure 26: Grad-CAM explanations on log-Mel spectrograms of audio content from videos for five randomly selected devices under the Native, WhatsApp, and YouTube scenarios.

Explainable Camera Model Identification

Experimental results

Table 16: Accuracy (%) comparison between the Grad-CAM-driven method and previous methods[30].

	Visual-ResNet-50			**Audio-ResNet-18**[30]			**Grad-CAM-driven Audio-ResNet-18**		
	Native	WhatsApp	YouTube	Native	WhatsApp	YouTube	Native	WhatsApp	YouTube
Fold 0	88.31	67.53	77.02	96.10	93.50	91.90	96.02	94.68	92.53
Fold 1	85.70	83.11	72.97	94.80	90.90	93.24	95.15	92.80	94.61
Fold 2	89.60	63.63	77.02	90.90	88.31	95.94	92.37	90.59	96.42
Fold 3	89.47	68.42	63.51	93.42	94.73	82.43	93.75	93.25	86.84
Fold 4	88.15	64.47	78.37	94.73	88.15	95.94	96.01	89.25	96.11
Mean $\pm$ StD	88.24 $\pm$ 1.40	69.43 $\pm$ 7.07	71.77 $\pm$ 5.44	93.99 $\pm$ 1.76	91.11 $\pm$ 2.66	91.89 $\pm$ 4.98	**94.66 $\pm$ 1.49**	**92.11 $\pm$ 2.11**	**93.30 $\pm$ 3.53**

Explainable Camera Model Identification

Experimental results

Table 17: Accuracy (%) comparison between the Product and Sum rules in previous and Grad-CAM-driven methods.

	Product Rule[30]			**Sum Rule**[30]			**Grad-CAM-driven Product Rule**			**Grad-CAM-driven Sum Rule**		
	Native	WhatsApp	YouTube	Native	WhatsApp	YouTube	Native	WhatsApp	YouTube	Native	WhatsApp	YouTube
Fold 0	97.40	94.80	95.94	97.40	96.10	94.59	98.67	96.12	96.91	97.91	97.82	96.02
Fold 1	97.40	94.80	94.59	96.10	96.10	93.24	98.01	95.91	94.11	97.89	97.05	94.01
Fold 2	98.70	93.50	95.94	97.40	90.90	97.29	98.92	93.21	94.98	98.08	92.21	95.51
Fold 3	97.36	94.73	90.54	94.73	97.36	86.48	98.73	95.11	91.32	96.51	95.88	90.09
Fold 4	97.36	86.84	95.94	96.05	88.15	97.29	98.22	90.18	96.38	97.12	90.92	96.77
Mean $\pm$ StD	97.64 $\pm$ 0.52	92.93 $\pm$ 3.08	95.59 $\pm$ 0.52	96.33 $\pm$ 0.99	93.72 $\pm$ 3.56	93.77 $\pm$ 3.97	**98.51 $\pm$ 0.34**	**94.11 $\pm$ 2.22**	**94.74 $\pm$ 1.98**	**97.50 $\pm$ 0.60**	**94.78 $\pm$ 2.72**	**94.48 $\pm$ 2.37**

- Let $\mathrm{T} = [\boldsymbol{t}^{(1)}, \ldots, \boldsymbol{t}^{(N)}] \in \mathbb{R}^{C \times N}$ be the matrix of target variables. The (c', n) element of $\boldsymbol{T}$ is denoted by $t_{c'}^{(n)}$. The target vector $\mathrm{t}^{(n)}$, that corresponds to the sample $\boldsymbol{x}_m^{(n)}$, adheres to the one-hot encoding scheme. That is, if $\boldsymbol{x}_m^{(n)}$ belongs to class $\mathcal{C}_{c'}$, the target vector $\boldsymbol{t}^{(n)}$ has zero elements except for the c'th element, which is set to one.
- Probability of assigning the input $\boldsymbol{x}_m^{(n)}$ to class $\mathcal{C}_{c'}$

$$y_{mc'}^{(n)} = \Pr(\mathcal{C}_{c'} \mid \boldsymbol{x}_m^{(n)} \, ; \boldsymbol{w}_m^{c',[L]}) = \Pr(\mathcal{C}_{c'} \mid \boldsymbol{x}_m^{(n)}), \tag{20}$$

- Activations of the last layer

$$\boldsymbol{a}_m^{(n)[L]} = \begin{bmatrix} \Pr(\mathcal{C}_1 \mid \boldsymbol{x}_m^{(n)} \, ; \boldsymbol{w}_m^{1,[L]}) \\ \Pr(\mathcal{C}_2 \mid \boldsymbol{x}_m^{(n)} \, ; \boldsymbol{w}_m^{2,[L]}) \\ \vdots \\ \Pr(\mathcal{C}_C \mid \boldsymbol{x}_m^{(n)} \, ; \boldsymbol{w}_m^{C,[L]}) \end{bmatrix} \in \mathbb{R}^C. \tag{21}$$

$$\mathcal{P}\left(\bigcup_{c'=1}^{C}\{\boldsymbol{w}_{m'}^{c'}\}_{m'=1}^{M}\right) = -\sum_{n=1}^{N}\sum_{c'=1}^{C}\sum_{m'=1}^{M} t_{c'}^{(n)} \cdot \ln \frac{\Pr(\mathcal{C}_{c'})^{1-M}\, y_{m'c'}^{(n)}}{\sum_{c''=1}^{C}\Pr(\mathcal{C}_{c''})^{1-M}\, y_{m'c''}^{(n)}}. \tag{22}$$

Online Fusion

Model updating

- The parameters of the model $\boldsymbol{w}_m^c$ can be updated in a manner of steepest descent by using the gradient

$$\nabla_{\boldsymbol{w}_m^c} \mathcal{P}\left(\bigcup_{c'=1}^{C} \{\boldsymbol{w}_{m'}^{c'}\}_{m'=1}^{M}\right) = -\sum_{n=1}^{N} \left(t_c^{(n)} - \tilde{y}_{mc}^{(n)}\right)\left(1 - y_{mc}^{(n)}\right)\ \boldsymbol{a}_m^{(n)[L-1]}. \tag{23}$$

- Each classifier can be trained separately.
- $\mathcal{P}\left(\bigcup_{c'=1}^{C} \{\boldsymbol{w}_{m'}^{c'}\}_{m'=1}^{M}\right)$ is convex w.r.t. $\boldsymbol{w}_m^c$, highlighting a desirable property of the online product rule loss function during the optimization procedure.

The class c is given by

$$c = \underset{c'=1}{\overset{C}{\operatorname{argmax}}} \Pr(\mathcal{C}_{c'})^{1-M} \prod_{m'=1}^{M} \Pr(\mathcal{C}_{c'} \mid \boldsymbol{x}_{m'}^{(n)}). \tag{24}$$

The product rule (24) can be used for multimodal fusion after the classifiers for each modality have been trained.

Online Fusion

Online product rule loss function enhanced with label smoothing

$$\breve{\mathcal{P}}\left(\bigcup_{c'=1}^{C}\{\boldsymbol{w}_{m'}^{c'}\}_{m'=1}^{M}\right) = (1-\epsilon)\mathcal{P}\left(\bigcup_{c'=1}^{C}\{\boldsymbol{w}_{m'}^{c'}\}_{m'=1}^{M}\right) - \frac{\epsilon}{C}\sum_{n=1}^{N}\sum_{c'=1}^{C}\sum_{m'=1}^{M}\ln\frac{\Pr(\mathcal{C}_{c'})^{1-M}\, y_{m'c'}^{(n)}}{\sum_{c''=1}^{C}\Pr(\mathcal{C}_{c''})^{1-M}\, y_{m'c''}^{(n)}}. \quad (25)$$

Table 18: Number of parameters for each deep learning model.

Model	Parameters (Millions)
SqueezeNet1_1[31]	0.741
MobileNetV3Small[32]	0.947
MobileNetV3Large[32]	3.320
ResNet-18[33]	11.194
ResNet-50[33]	23.578

[31] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, "SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size," *arXiv:1602.07360*, 2016.

[32] A. Howard, M. Sandler, G. Chu, L. -C. Chen, B. Chen, M. Tan, W. Wang, Y. Zhu, R. Pang, V. Vasudevan, et al., "Searching for MobileNetV3," in *Proc. of the IEEE Conference on Computer Vision and Pattern Pecognition*, pp. 1314–1324, 2019.

[33] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, 2016.

Online Fusion

Mean SDI accuracy

Table 19: The mean SDI accuracy (%) and the corresponding StD in each fold, after applying online fusion to tune the MobileNetV3Large networks used for classifying visual and audio content. Late fusion is performed in each fold between these two models, with the results reported under the column Late Fusion.

	Fine-Tuned Models						**Late Fusion (Visual Net/Audio Net)**		
	Visual Net (MobileNetV3Large)			**Audio Net (MobileNetV3Large)**			**MobileNetV3Large/MobileNetV3Large**		
	Native	WhatsApp	YouTube	Native	WhatsApp	YouTube	Native	WhatsApp	YouTube
Fold 0	80.38/85.37	69.15/79.20	61.56/69.42	91.42/96.75	91.29/96.80	87.52/91.74	93.67/95.93	93.83/99.20	87.69/95.86
Fold 1	85.41/93.65	72.92/82.40	55.21/74.38	92.73/96.83	91.19/97.60	83.93/92.56	96.65/100.0	93.71/97.60	81.72/92.56
Fold 2	81.50/86.40	67.22/78.40	60.07/66.94	91.63/96.80	91.86/95.20	81.86/90.91	93.49/95.20	92.23/94.40	80.28/90.90
Fold 3	79.80/86.51	63.89/76.80	63.48/69.17	90.75/95.24	86.63/92.00	86.13/91.67	94.36/95.23	90.49/95.20	88.12/95.00
Fold 4	76.82/81.60	64.35/74.40	68.68/70.83	88.67/95.20	91.48/95.20	87.34/92.50	91.23/93.60	92.30/96.00	92.15/92.50
Mean ± StD	80.78/86.70 ± 2.78/3.90	67.50/78.24 ± 3.32/2.64	61.80/70.14 ± 4.39/2.45	91.04/96.16 ± 1.34/0.77	90.49/95.36 ± 1.94/1.92	85.35/91.87 ± 2.16/0.60	93.88/95.99 ± 1.73/2.14	92.51/96.48 ± 1.21/1.72	85.99/93.96 ± 4.38/1.91

Concluding Remarks

- AEGIS dataset including video and ground truth ENF will be released.
- Device identification can be cast as a learning problem within a unified hypergraph framework representing model and brand information.
- Other research avenues include continual learning and federated learning to address distributed learning and promote person sensitive data preservation.
- Niche applications will be pursued, e.g., Social Network Identification. Are there suitable fingerprints to be exploited for detecting text generated by ChatGPT in a passive mode?

Acknowledgement

This research was supported by the Hellenic Foundation for Research and Innovation (H.F.R.I.) under the "2nd Call for H.F.R.I. Research Projects to support Faculty Members & Researchers" (Project Number: 3888).

Acknowledgement

This research was supported by the Hellenic Foundation for Research and Innovation (H.F.R.I.) under the "2nd Call for H.F.R.I. Research Projects to support Faculty Members & Researchers" (Project Number: 3888).

Thank you!

Any Questions?