Scattering Wavelet Hash Fingerprints for Musical Audio Recognition

Evren Kanalici, Gokhan Bilgin

Abstract: ***Fingerprint design is the cornerstone of the audio recognition systems in which aims robustness and fast retrieval. Short-term Fourier transform and Mel-spectral representations are common for the task in mind, however these extraction methods suffer from being unstable and having limited spectral-spatial resolution. Scattering wavelet transform (SWT) provides another approach to these limitations by recovering information loss, while ensuring translation invariance and stability.***

We propose a two-stage feature extraction framework using SWT coupled with deep Siamese hashing model for musical audio recognition. Similarity-preserving hashes are the final fingerprints and in the projected embedding space, similarity is defined by a distance metric. Hashing model is trained by roughly aligned and non-matching audio snippets to model musical audio data via two-layer scattering spectrum. Our proposed framework provides competitive performance results to identify audio signals superimposed with environmental noise which can be modeled as real-world obstacles for music recognition. With a very compact storage footprint (256 bytes/sec.), we achieve 98.2% ROC AUC score on GTZAN dataset.

Index Terms**: Audio fingerprinting, CNNs, Scattering wavelet transform, Siamese networks, Embedding hash models.**

I. INTRODUCTION

An audio fingerprint is a content-based compact signature that represents audio signal. Fingerprinting/hashing is an engineering task which includes fingerprint design, similarity metric and matching search for verification and recognition/identification of data interested. Real-world applications for audio fingerprinting varies from query-by-example music recognition, audio labeling, content-based integrity verification to copyright detection systems. For instance, popular online music labeling systems are available including Shazam [1] [2], SoundHound [3] and Google Sound Search [4] with client-server architecture. A general architecture for a fingerprinting & recognition systems is depicted in Fig. 1.

Considering the tradeoff of system resources of real-world scenarios, precision/recall and response time requirements, what the research is mainly focused on is design of fingerprints that are essential i.e. being robust to various degradations, alignment problems in matching step, feature ex- traction/calculation complexity and compactness for fast retrieval. An audio signal may undergo many kind of distortions like additional background noise, reverberations, pitch shifting, interference in transmission, quantization and/or compression artifacts (i.e. GSM or MP3). The designed system should tackle with these obstacles.

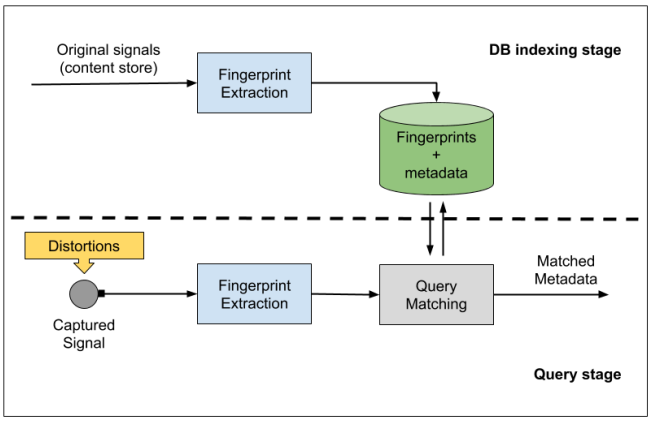

Figure 1: General architecture of an audio finger-printing system

Approaching to audio fingerprint problem in spectral domain is common, which benefits from image processing techniques after the change of domain. Haitsma and Kalker [5] propose fingerprints using Bark Frequency Cepstrum Coefficients (BFCC). To ensure having time-alignment neutral features, highly overlapping audio frames are considered and Hamming distance is used for comparison. Ke *et al*. [6] insight is, 1D audio signals can be processed as conventional images when transformed into time-frequency representations. They train AdaBoost classifiers based on box-filters, then the concatenated classifiers' output from sub-rectangular regions of the spectrogram image is used as fingerprints. Wang *et al*. [1] use local spectral energy peaks of time-frequency points with neighborhood of L2 distance. The fingerprints are sparse and its claimed they are discriminative enough and robust noise artifacts. Baluja and Covell applied methods from computer vision [7] to extract haar-wavelet fingerprints with short time support, then quantized values can be indexed at large scale using locality-sensitive hashing [8]. In Gfeller *et al*. [9] work, spectrogram coefficients are fed to convolutions neural network to learn hash embeddings. Their neural network fingerprinter generates 96-dimensional embeddings at a rate of one second after trained with triplet loss function. Then for identification, L2 distance combined with adaptive scoring is used.

In this paper we explore a two-stage feature extraction method combining scattering wavelet transform (SWT) and deep embedding hash model to generate final fingerprints. SWT generates contractive representation of signals and provides Lipschitz continuity to deformations

Revised Manuscript Received on June 7, 2019.

Evren Kanalici, Dpt. of Computer Engineering, Yildiz Technical University, Istanbul, Turkey.

Gokhan Bilgin, Dpt. of Computer Engineering, Yildiz Technical University, Istanbul, Turkey.

[10]. That is the distance between the transforms of the degraded and the original signals are bounded for a group of deformations satisfying necessary conditions. Then we build our deep hashing network based on combination of first-order and second-order scattering coefficients to model musical data efficiently. We experiment with naive identification retrieval methods not to be influenced by database precision artifacts.

In following section (Section-2), we explain scattering transform and our deep embedding model which takes inputs of scattering coefficients. In Section-3 and Section-4, evaluations and experimental results of our framework is presented for musical audio recognition.

II. METHODOLOGY

A. Scattering Wavelet Transform

Wavelet scattering transform, presented by Mallat [11], is based on arguments of stability under dilation and transposition. With insights that short-term Fourier transform (STFT) is unstable, Mel-frequency representation provides stability by band-pass averaging:

(1.1)

$$Mx(t,\lambda) = \frac{1}{2\pi}\int |\hat{x}(w,t)||\hat{\psi}_{\lambda}(w)|^2 dw$$

where $\hat{\psi}_{\lambda}$ is a band-pass filter at Mel-frequency λ.

The deformation stability comes at the cost of information loss in high frequencies that are averaged and in short- term windowed temporal structure. SWT provides both Mel-frequency and modulation features by recovering averaged lost information. For a given signal x, complex wavelet transform Wx is defined as convolutions with a averaging operator (low-pass filter) ϕ and higher frequency wavelets ψ_{λ_1}:

$$Wx(t,\lambda) = \left(x \star \phi(t), x \star \psi_{\lambda_1}(t)\right)_{t\in\mathbb{R},\lambda_1\in\Lambda_1}$$

(1.2)

The wavelet power spectrum (modulus operator $|\bullet|$, being a non-linear map, is translation-invariant and is stable to deformations as argued in [12]) extracts time-windowed envelopes at different resolutions (Eq. 3). Modulus discards phase information but retains sufficient information in the nature of wavelet transform being redundant.

(1.3)

$$|W|x = \left(x \star \phi(t), |x \star \psi_{\lambda_1}(t)|\right)_{t\in\mathbb{R},\lambda_1\in\Lambda_1}$$

Whereas zeroth-order coefficients $S_0 = x \star \phi(t)$ is locally invariant to translation, the high-frequency information is lost by time-averaging ϕ. Although lost information is obtained with modulus of wavelet coefficients $|x \star \psi_{\lambda_1}|$, yet these coefficients are not time-shift invariant. Provided that local time-shift invariance is obtained by time averaging as in S_0, applying averaging again gives first-order of scattering coefficients:

(1.4)

$$S_1 x(t,\lambda_1) = |x \star \psi_{\lambda_1}| * \phi(t)$$

If the wavelets ψ_{λ_1} have the frequency resolution as the standard mel-scale i.e. $\lambda(f) = 1127\, ln(1+ f/700)$, the first-order coefficient $S_1 x$ approximate the mel-spectrogram as it is shown in [12]. What's more, scattering transform enables to recover lost information in higher-order components by passing the modulus coefficients $|x \star \psi_{\lambda_1}|$ through a bank of higher frequency wavelets ψ_{λ_2} (Eq.5), and then applying time-averaging operator gives second-order coefficients (Eq.6):

(1.5)

$$|W_2||x \star \psi_{\lambda_1}| = \left(|x \star \psi_{\lambda_1}| \star \phi, ||x \star \psi_{\lambda_1}| \star \psi_{\lambda_2}|\right)_{\lambda_2\in\Lambda_2}$$

(1.6)

$$S_2 x(t,\lambda_1,\lambda_2) = ||x \star \psi_{\lambda_1}| \star \psi_{\lambda_2}| \star \phi(t)$$

Repeatedly applying modulus for stability to deformations (diffeomorphism) and time-averaging for translation invariance in layer-wise fashion leads to the scattering spectrum as depicted in Fig. 2. Although 2-layered scattering is successful enough for most of audio signal applications capturing mel-spectrogram and modulation features, noted also coefficients goes to zero as higher layer transforms are applied. In practical level i.e. for featurs extraction, parameters for support of averaging low-pass filter $\phi(t)$ (*invariance-scale*) and filter-bank quality per each order may be considered.

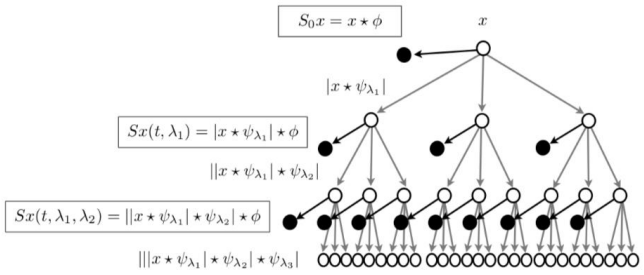

Figure 2: Scattering wavelet spectrum [12]

B. Fingerprint Hashing Model

Supervised learning-based hashing models are used for compact representation of domain specific input for various tasks including classification, recognition and verification. The trained model provides a mapping from input space to an embedding space. Embeddings learned in such way can be used as features vectors. After a non-linear projection of embedding space is modeled, a distance metric (also the training loss of the model) may be defined, then recognition task can be approached as a simple k-nearest neighbors (kNN) clustering task if the embedding space is Euclidian.

We build our fingerprint hashing model by the combination of stack of convolutional layers (CNNs), fully-connected layers and divide-and-encode block

[13] on top. CNNs can model spectral correlations of our first-order scattering coefficients S_1. While second-order scattering $S_2x(t,\lambda_1,\lambda_2)$ which is the decomposition of modulation features in each sub-band of the first-order filter-bank ($|x \star \psi_{\lambda_1}|$) preserves the local information for given sub-band λ_1, being more sparse with few non-close coefficients, are fed to fully-connected layer. Our model is shown in Fig. 3.

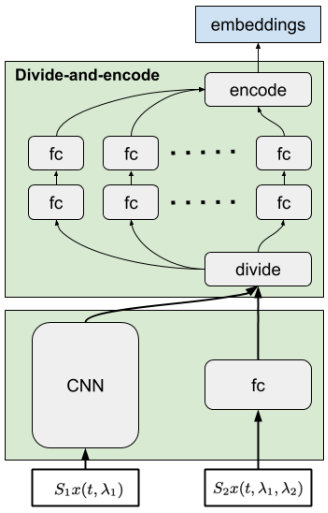

Figure 3: Overview of embedding hashing model

Two-level of divide-and-encode block splits the intermediate representation before combining into the final embed- dings. The idea of choosing divide-and-encode block is to reduce the redundancy among intermediate hashes as suggested in [13]. Except final divide-and-encode layer all layers use Parametric ReLU activation function and batch normalization. We trained the network as Siamese network of audio snippet similarities using triplet loss function [14]. The constraints for audio snippets to be considered as same are they should be from same audio file and their starting position should differ only couple of hundred milliseconds. Siamese networks with L2 loss functions are useful to learn map- pings from input data to a compact Euclidean space where distances correspond to a measure of similarity. For an embedding mapping f, triplet loss function is given in Eq.7:

$$L^+ = \left\| f(x_a) - f(x_p) \right\|^2$$
$$L^- = \left\| f(x_a) - f(x_n) \right\|^2 \qquad (1.7)$$
$$L_{triplet}(x_a, x_p, x_n) = max(0, margin + L^+ - L^-)$$

where (x_a, x_p, x_n) are anchor, positive and negative samples accordingly.

III. EVALUATION

We used GTZAN dataset [15] [16] for our experiments. In training stage of our embedding hash model, various additional noise applied to audio signals to preserve similarity for degradations. In evaluation step, selected environmental and artificial noises are applied with adjusted SNR values as explain in next section. For implementation, to build our embedding hash model Pytorch framework [17], and for scattering transform of audio signals Kymatio framework [18] are used.

GTZAN dataset includes 1000 audio files from 10 musical genres each 30-seconds long and with sampling rate 22050Hz. We prepared 10-seconds longs audio snippets down- sampled to Fs=16kHZ, having total number of 3K snippets, then randomly split at ratio of (0.8, 0.2) for training and test accordingly. For scattering transform, after our experiments we chose the support of averaging filter ϕ to be 2^9 samples, giving an invariance-scale of ~32ms ($2^9 / Fs$), and the number of first-order wavelets per octave to be 8. Feature vectors consist of one-second long scattering coefficients resulting 31×299 dimensional sub-rectangles (i.e. $1/(2^9/Fs) = 31$). We define a *neighbor distance* that equals to $370ms$, to label audio segments considered as same or not by their starting position while training. Also, while we are training our model, we use strides equal to invariance-scale duration for each audio snippets, whereas database indexing stage use features per every one-second without overlapping. General overview of audio signal fingerprint scheme is depicted in Fig. 4.

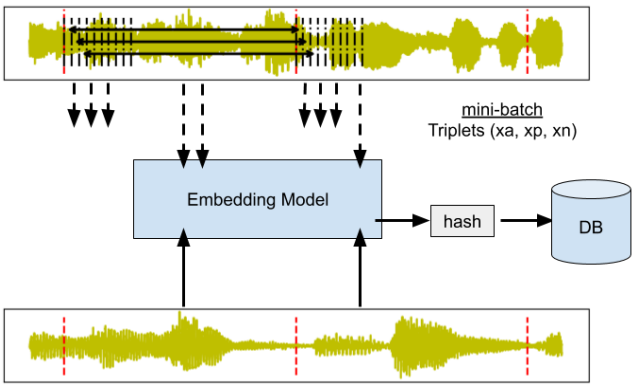

Figure 4: Audio signal fingerprinting scheme. Embedding model training (top), Database indexing (bottom).

To prevent biased learning, first-order S_1 scattering coefficients are mean and variance normalized, whereas for second-order S_2 coefficient values, first *scattering transfer* normalization [12], later mean-normalization is applied. While training, for each mini-batch online triplet selection strategies are executed to prevent poor training of the network [14] [19]. Each mini-batch contains all possible positive triplets in the neighborhood distance while negatives are selected using *semi-hard exemplars*, that are further away from the anchor than positive exemplar, but inside the margin. We prepared a noisy variant (superimposed with various environmental noise) of the dataset with **same alignment**, and in our scenario, anchor and positive samples regard to clean and noisy audio segments within the defined neighborhood accordingly.

IV. EXPERIMENTS

After training the embedding hash model, the content database is built

from clean test samples using model as the feature extractor. Content database consist of test split of GTZAN having #600 10-seconds long snippets. Each snippet is represented by features per every one-second without overlapping. Retrieval is executed naively by one-by-one comparison not to be influenced by database precision artifacts. One-second long segments are randomly sampled from query snippets with *arbitrary alignments* and features are extracted using the same embedding model. For each compared features against database, L2 distances are calculated and the features which regards top-K smallest distance values are selected as candidates.

First we query the database with existing clean samples and calculate the top-K=1 mean score for multiple models. For each 10-seconds long snippets #10 randomly sampled segments are selected and their features are compared against the database. If the first result is a match (i.e. same song within the neighborhood offset) we increment the score. The reason of this experiments is to evaluate how margin value and hash dimension affect the retrieval probability of alignment and shift-invariance without any additional noise. Results are given in Table-1. As it can be seen best similarity preserving embeddings are obtained for 96-dimensional vectors and for margin=1.0, although after 32-dimensional hash model, having more dimension doesn't seem to have great impact.

Table 1: Mean top-K=1 positive retrieval probability

Hash Dim.	margin=0.2	margin=1.0	margin=2.0
8	0.8622	0.8731	0.8589
16	0.9266	0.9185	0.9011
32	0.9310	0.9343	0.9297
64	0.9404	0.9358	0.9263
96	0.9354	**0.9411**	0.9297

By selecting best margin=1.0 value from Table-1, we calculate the same scores for noisy samples with various signal-to-noise-ratio (SNR) values. Experimented additional noise include (1) chatting people, in down-town streets, (2) noisy wind sound effect, (3) sound recorded inside a window of rainy day and (4) artificial synthetic glitch noise, all retrieved from freesound.org [20] website. Results are given in Table-2.

We can see the dimensionality factor of the embeddings on discrimination in Table-2 clearly, for hash dimension equal to 96 we can retrieve positive samples with about 93% recall from a sparse database having features per only every one-second of audio signals with a very compact representation ($96 \times 4 = 384$ bytes/sec.). Also noted, the positive retrieval score for noise type (3) is poor having less than 50% for SNR below 6.

Table 2: Mean top-K=1 positive retrieval probability for noisy samples

Hash Dim.	SNR=9	SNR=6	SNR=3	SNR=0
(1) group-people-chatting-city				
16	0.8468	0.7711	0.6327	0.4100
32	0.8685	0.8264	0.7252	0.5281
64	0.8958	0.8608	0.7737	0.6206
96	0.9064	0.8770	0.8020	0.6625
(2) wind-on-microphone				
16	0.8581	0.7947	0.6618	0.4522
32	0.9025	0.8854	0.8341	0.7166
64	0.9245	0.9010	0.8833	0.8168
96	**0.9312**	0.9197	0.8884	0.8333
(3) raindrops-on-the-windows				
16	0.6497	0.4625	0.2975	0.1412
32	0.7479	0.5954	0.3816	0.2006
64	0.8052	0.6697	0.4779	0.2737
96	0.8397	0.7279	0.5722	0.3677
(4) artificial-synthetic-noise				
16	0.8120	0.6897	0.5008	0.2768
32	0.8658	0.8102	0.6737	0.4710
64	0.8966	0.8502	0.7845	0.6416
96	0.9056	0.8816	0.8166	0.6912

Lastly, a concrete recognition search is done against our indexed DB with top-K=20, i.e. each fingerprint can propose up to twenty potential matches. Query set includes both positive and negative samples and query snippets are superimposed with randomly selected noise type with SNR values varying from 0 to 3 (i.e. between half and same energy of noise applied). We execute the retrieval process with a sequence of snippets of audio with **random alignment** but with **increasing starting offsets**. The matches from database includes time-offset metadata, so we define an *adaptive scoring* method for sequential retrieval of an audio signal using following temporal constraints as a basis of dynamic-programming (similar to constraints in [8]) to decide whether proposed candidates are a match or not:

1. First, we define an anchor (initial) match for all of top- K candidates.
2. Second, for the tail of the anchor match, we define following temporal constraints:
   - Do not allow temporal backtracking of matched features.
   - Allow only matches the within the defined temporal cone for anchor match.
3. Promote the longest match sequence as the candidate.

Fig. 5 shows Receiver operating characteristic (ROC) curve of the retrieval performance for various dimensional embeddings.

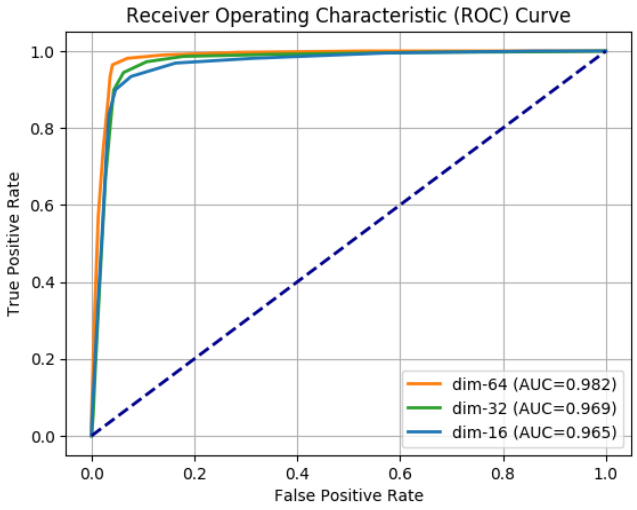

Figure 5: Retrieval performance with adaptive query scoring against hash dimensions.

V. CONCLUSION AND FUTURE WORK

With an adaptive scoring scheme, we can retrieve **0.982** ROC score for **64-dimensional** hash fingerprints. Our database is sparse, indexing only 64 floating-values (i.e. $64 \times 4 = 256$ bytes) per second for indexed audio signals without over-lapping. Also limiting the hash dimension doesn't seem to have great impact as we suspect that because the database size is limited. For 16-dimensional hashes (64 bytes/sec.) the ROC score is above 95% (see Fig. 5).

In this paper we tried to tackle most common obstacle for audio identification from the end-user perspective that is musical audio tampered with environmental noise. To be more comprehensive other group of degradation should be considered, such as how sampling rate changes effect the output our framework or could using features of long durations (i.e. one second) compensate linear-speed modification of audio signal should be answered. If the latter is not, keeping minimal duration of features vs. storage footprint trade-off should be adjusted carefully.

Furthermore, firstly we should apply our technique to larger datasets. And secondly, we should evaluate the adaptation of network model changes using harmonic embeddings. Harmonic embeddings is a technique of adapting newly designed models to be able to improve verification accuracy while maintaining compatibility to less accurate embeddings of initial model [14]. We plan to work on these topics in our followings works.

REFERENCES

1. A. Wang et al. An industrial strength audio search algorithm. In Ismir, volume 2003, pages 7–13. Washington, DC, 2003.
2. A. Wang. The shazam music recognition service. Communications of the ACM, 49(8):44–48, 2006.
3. A. S. Master, T. P. Stonehocker, B. J. Levitt, J. Huang, and K. Mohajer. Systems and methods for sound recognition, Mar. 8 2016. US Patent 9,280,598.
4. Google sound search. https://support.google.com/googleplaymusic/answer/2913276.
5. J. Haitsma and T. Kalker. A highly robust audio fingerprinting system. In Ismir, volume 2002, pages 107–115, 2002.
6. Y. Ke, D. Hoiem, and R. Sukthankar. Computer vision for music identification. In 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), volume 1, pages 597–604. IEEE, 2005.
7. J. Z. Wang, G. Wiederhold, O. Firschein, and S. X. Wei. Wavelet-based image indexing techniques with partial sketch retrieval capability. In Proceedings of ADL'97 Forum on Research and Technology. Advances in Digital Libraries, pages 13–24. IEEE, 1997.
8. S. Baluja and M. Covell. Audio fingerprinting: Combining computer vision & data stream processing. In 2007 IEEE International Conference on Acoustics, Speech and Signal Processing-ICASSP'07, volume 2, pages II–213. IEEE, 2007.
9. B. Gfeller, R. Guo, K. Kilgour, S. Kumar, J. Lyon, J. Odell, M. Ritter, D. Roblek, M. Sharifi, M. Velimirovi´c, et al. Now playing: Continuous low-power music recognition. arXiv preprint arXiv:1711.10958, 2017.
10. J. Bruna and S. Mallat. Invariant scattering convolution networks. IEEE transactions on pattern analysis and machine intelligence, 35(8):1872–1886, 2013.
11. S. Mallat. Group invariant scattering. Communications on Pure and Applied Mathematics, 65(10):1331–1398, 2012.
12. J. Anden and S. Mallat. Deep scattering spectrum. IEEE Transactions on Signal Processing, 62(16):4114–4128, 2014.
13. H. Lai, Y. Pan, Y. Liu, and S. Yan. Simultaneous feature learning and hash coding with deep neural networks. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 3270–3278, 2015.
14. F. Schroff, D. Kalenichenko, and J. Philbin. Facenet: A unified embedding for face recognition and clustering. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 815–823, 2015.
15. G. Tzanetakis and P. Cook. Musical genre classification of audio signals. IEEE Transactions on speech and audio processing, 10(5):293–302, 2002.
16. G. Tzanetakis and P. Cook. Gtzan genre collection. http://marsyas.info/downloads/datasets.html, 2002.
17. A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer. Automatic differentiation in pytorch. 2017.
18. M. Andreux, T. Angles, G. Exarchakis, R. Leonarduzzi, G. Rochette, L. Thiry, J. Zarka, S. Mallat, E. Belilovsky, J. Bruna, et al. Kymatio: Scattering transforms in python. arXiv preprint arXiv:1812.11214, 2018.
19. A. Hermans, L. Beyer, and B. Leibe. In defense of the triplet loss for person re-identification. arXiv preprint arXiv:1703.07737, 2017
20. F. Font, G. Roma, and X. Serra. Freesound technical demo. In Proceedings of the 21st ACM international conference on Multimedia, pages 411–412. ACM, 2013.

AUTHORS PROFILE

Evren Kanalici received his BSc degree in Computer Engineering from the Yildiz Technical University (YTU), Istanbul, Turkey, in 2012. He is currently working toward the M.Sc. degree in the Computer Engineering, YTU. His main research interests are signal processing, computer vision, music information retrieval, pattern recognition and machine learning. He is a member of the Signal and Image Processing Laboratory (SIMPLAB), Department of Computer Engineering, YTU.

Gokhan Bilgin received his BSc, MSc, and PhD degrees in Electronics and Telecommunication Engineering from Yildiz Technical University (YTU), Istanbul, Turkey, in 1999, 2003, and 2009, respectively. He worked as a postdoctoral researcher at Indiana University-Purdue University at Indianapolis, USA. Currently, he is working as an Associate Professor at the Department of Computer Engineering and as the head of the Signal and Image Processing Laboratory (SIMPLAB) in YTU. His research interests are in the areas of image and signal processing, machine learning and pattern recognition with applications to biomedical engineering and remote sensing.